AIX 7.2 változat

Biztonság

AIX 7.2 változat

Biztonság

Megjegyzés

Az információk és a tárgyalt termék használatba vétele előtt olvassa el a “Nyilatkozatok” oldalszám: 497 helyen található általános tájékoztatást.

This edition applies to AIX Version 7.2 and to all subsequent releases and modifications until otherwise indicated in new editions.

Tartalom

Tudnivalók a dokumentumról

Ez a témakör-gyűjtemény a rendszeradminisztrátorok számára az információk teljes körét biztosítja a fájlokról, rendszerekről és hálózati biztonságról. A témakör-gyűjtemény olyan feladatok végrehajtását írja le, mint a rendszer megerősítése, az engedélyek módosítása, a hitelesítési módszerek beállítása és a Common Criteria Security Evaluation szolgáltatások beállítása. Ez a témakör-gyűjtemény az operációs rendszerrel együtt szállított dokumentációs CD-n is elérhető.

Kiemelések

A dokumentum az alábbi kiemelési megállapodásokat használja:

Félkövér	Parancsokat, szubrutinokat, kulcsszavakat, fájlokat, szerkezeteket, alkönyvtárakat és más olyan elemeket jelöl, amelyeknek nevét a rendszer előre meghatározza. Felhasználó által kijelölt grafikus objektumokat, például nyomógombokat, címkéket és ikonokat is azonosít.
Dőlt	Azokat a paramétereket jelöli, amelyeknek nevét vagy értékét a felhasználó adja meg.
`Rögzített szélességű`	Adott adatértékekre vonatkozó példákat, ténylegesen megjelenő mintaszövegeket, programozók által felhasználható kódrészleteket, rendszerüzeneteket, vagy beírandó információkat jelöl.

Kis- és nagybetűk megkülönböztetése AIX rendszeren

Az AIX operációs rendszerben a kis- és nagybetűk mindenhol meg vannak különböztetve. A fájlokat például az **ls** paranccsal listázhatja ki. Ha beírja az LS parancsot, akkor a rendszer azt a választ adja, hogy a parancs nem található. Hasonlóképp, a **FILEA**, a **FiLea** és a **filea** különböző fájlnevek, még akkor is, ha ugyanabban a könyvtárban vannak. A nem kívánt műveletek végrehajtásának elkerülése érdekében mindig ügyeljen a kis- és nagybetűk helyes használatára.

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

Biztonság

Az AIX operációs rendszer lehetővé teszi olyan feladatok végrehajtását, mint például a rendszer megerősítése, az engedélyek módosítása, a hitelesítési módszerek beállítása és a Common Criteria Security Evaluation szolgáltatások beállítása. Ez a témakör-gyűjtemény az operációs rendszerrel együtt szállított dokumentációs CD-n is elérhető.

Kapcsolódó tájékoztatás:

- Számítógépes vészhelyzeti csoport a Carnegie Mellon egyetemen (CERT)
- Forum of Incident Response and Security Teams (FIRST)
- Center for Education and Research in Information Assurance and Security (CERIAS)

Újdonságok a Biztonság témakörben

Itt a Biztonság témakör új és jelentősen megváltozott információiról olvashat.

Hogyan találom meg az újdonságokat és a módosításokat?

A PDF fájlban az új és módosított információk mellett függőleges vonal (|) látható a bal oldalon.

2017. január

A témakörben végzett frissítések összefoglaló információi:

- Hozzáadásra kerültek információk a megfigyelési eseményekkel kapcsolatban a “Megfigyelési események” oldalszám: 138 témakörben.
- Hozzáadásra kerültek információk az OpenSSH képekkel kapcsolatban az “OpenSSH telepítőkészletek” oldalszám: 198 témakörben.

Az alap operációs rendszer biztonságossá tétele

Az alap operációs rendszer biztonságossá tétele információkat nyújt arról, hogyan kell védeni a rendszert, tekintet nélkül a hálózati csatlakozásra.

A fejezetből megtudhatja, hogyan telepíthető a rendszer bekapcsolt biztonsági lehetőségekkel, és hogyan védhető meg az AIX a jogosulatlan felhasználói hozzáférésekkel szemben.

Biztonságos rendszer telepítése és beállítása

Az AIX biztonságos telepítésekor és beállításakor számos szempontot kell figyelembe venni.

Megbízható számítástechnikai alapkörnyezet

Egy adott program megbízhatóságát a rendszeradminisztrátor határozza meg. E meghatározás során számításba kell venni a rendszer információs erőforrásainak értékét és el kell dönteni, milyen mértékű megbízhatóság szükséges egy jogosultságokkal rendelkező program telepítéséhez.

A Megbízható számítástechnikai alapkörnyezet (TCB) a rendszer azon része, amely a teljes rendszerre kiterjedő információs biztonsági irányelvek betartásáért felel. A TCB telepítésével és használatával meghatározható a felhasználók hozzáférése a felhasználók és a TCB biztonságos kommunikációját megvalósító megbízható kommunikációs útvonalhoz. A TCB funkciói csak az operációs rendszer telepítése után engedélyezhetők. Ha egy már telepített gépre kívánja a TCB-t telepíteni, akkor először egy megőrző típusú telepítést kell végrehajtania. A TCB engedélyezése után használható a megbízható héjprogram, a megbízható folyamatok, valamint a Biztonságos Figyelem Kulcs (SAK).

A TCB ellenőrzése:

Az operációs rendszer biztonsága veszélybe kerül, ha a Megbízható számítástechnikai alapkörnyezet (TCB) fájljai nincsenek megfelelően védve, vagy ha a konfigurációs fájlok nem biztonságos értékeket tartalmaznak.

A **tcbck** paranccsal figyelhető a Megbízható számítástechnikai alapkörnyezet biztonsági állapota. A **tcbck** parancs ezeket az információkat figyeli meg az /etc/security/sysck.cfg fájl kiolvasásával. Ebben a fájlban tárolódik az összes TCB fájl, konfigurációs fájl és megbízható parancs leírása.

Az /etc/security/sysck.cfg fájl nem offline, tehát egy cracker módosíthatja. Gondosan ügyeljen arra, hogy a TCB minden egyes frissítése után készítsen egy offline, csak olvasható másolatot erről a fájlról. Ezenfelül bármilyen ellenőrzés elvégzése előtt másolja a fájlt az archív médiáról a lemezre.

A sysck.cfg fájl szerkezete:

A **tcbck** parancs beolvassa a /etc/security/sysck.cfg fájlt annak meghatározásához, hogy mely fájlokat kell ellenőrizni. A rendszer minden egyes megbízható programját az /etc/security/sysck.cfg fájl egy szakasza írja le.

Minden egyes szakasz az alábbi jellemzőkkel bír:

Attribútum	Leírás
acl	A fájl hozzáférés felügyeleti listáját reprezentáló szöveges karaktersorozat. Ugyanabban a formátumban kell, hogy legyen, mint az **aclget** parancs kimenete. Ha ez nem egyezik meg a fájl tényleges hozzáférés felügyeleti listájával (ACL), akkor a **sysck** parancs erre az értékre állítja be a fájl ACL-jét az **aclput** paranccsal. **Megjegyzés:** A SUID, SGID és SVTX attribútumoknak meg kell egyezniük a módban megadottakkal, amennyiben léteznek.
class	Egy fájlcsoport neve. Ez az attribútum lehetővé teszi ugyanazon osztályba tartozó több fájl ellenőrzését a **tcbck** parancs egyetlen argumentumával. Egynél több osztály is megadható, a neveiket vesszővel kell elválasztani.
group	A fájlcsoport csoportazonosítója vagy neve. Ha ez nem egyezik meg a fájl csoportjával, akkor a **tcbck** parancs erre az értékre állítja be a fájl csoportját.
links	Erre a fájlra hivatkozó elérési utak vesszővel elválasztott listája. Ha a listában megadott bármely elérési út nem hivatkozik erre a fájlra, a **tcbck** parancs létrehozza a hivatkozást. Ha a **tcbck** parancsot a *tree* paraméter nélkül használja, akkor kiír egy üzenetet, hogy extra hivatkozások találhatók, de nem határozza meg azok nevét. Ha a **tcbck** parancsot a *tree* paraméterrel együtt használja, akkor kiírja az adott fájlra hivatkozó további elérési utakat is.
mode	Értékek vesszővel elválasztott listája. Az engedélyezett értékek a SUID, SGID, SVTX és a TCB. A fájlengedélyek a legutolsóként - akár oktálisan, akár egy kilenckarakteres sorozatként - megadott értékkel kell, hogy megegyezzenek. A 755 vagy rwxr-xr-x például egyaránt érvényes fájljogosultságok. Ha ez nem egyezik meg a fájl hatályos módjával, akkor a **tcbck** parancs erre az értékre állítja be a fájl módját.
owner	A fájltulajdonos felhasználói azonosítója vagy neve. Ha ez nem egyezik meg a fájl tulajdonosával, akkor a **tcbck** parancs erre az értékre állítja be a fájl tulajdonosát.
program	Értékek vesszővel elválasztott listája. Az első érték egy ellenőrzőprogram elérési útja. A további értékek átadásra kerülnek a programnak futtatáskor argumentumként. **Megjegyzés:** Az első argumentum mindig a *-y*, *-n*, *-p* vagy a *-t* kapcsoló egyike attól függően, hogy a **tcbck** parancs melyik kapcsolóval lett futtatva.
source	Annak a fájlnak a neve, amelyből ez a forrásfájl átmásolandó ellenőrzés előtt. Ha az érték kitöltetlen és a fájl egy szabályos fájl, könyvtár, vagy névvel ellátott csővezeték, akkor a fájl egy új, üres verziója kerül létrehozásra, ha még nem létezik. Eszközfájlok esetén egy új speciális fájl kerül létrehozásra az ugyanolyan típusú eszközhöz.
symlinks	Erre a fájlra szimbolikusan hivatkozó elérési utak vesszővel elválasztott listája. Ha a listában megadott bármely elérési út nem szimbolikus hivatkozás erre a fájlra, akkor a **tcbck** parancs létrehozza a szimbolikus hivatkozást. Ha a **tcbck** parancsot a *tree* argumentummal együtt használja, akkor a kiírja az erre a fájlra szimbolikusan hivatkozó további elérési utakat is.

Ha az /etc/security/sysck.cfg fájl egy szakasza nem határozza meg valamely jellemzőt, akkor a hozzá tartozó ellenőrzés nem kerül elvégzésre.

A tcbck parancs használata:

A **tcbck** parancs az alábbiakat biztosítja: a biztonsággal kapcsolatos fájl megfelelő telepítését; a fájlrendszerfa ne tartalmazzon olyan fájlokat, amelyek nyilvánvalóan veszélyeztetik a rendszer biztonságát; a megbízható fájlok frissítését, hozzáadását vagy törlését.

A **tcbck** parancs jellemzően az alábbi feladatokra használható:

- A biztonsággal kapcsolatos fájlok megfelelő telepítésének garantálása
- Annak biztosítása, hogy a fájlrendszer nem tartalmaz a rendszer biztonságát világosan megsértő fájlokat
- Megbízható fájlok frissítése, hozzáadása vagy törlése

A **tcbck** parancs az alábbi módokon használható:

- Normális használat
  - Beavatkozást nem igénylő módban, a rendszer inicializálásakor
  - A **cron** paranccsal időzítve
- Interaktív használat
  - Egyes fájlok és fájlosztályok ellenőrzése
- Paranoid használat
  - Tárolja el a sysck.cfg fájlt offline módon, és időről időre állítsa vissza a gép ellenőrzése előtt

Bár kriptográfiai értelemben nem biztonságos, a TCB a **sum** parancsát használja az ellenőrző összegek kiszámításához. A TCB adatbázis beállítható kézzel más ellenőrzőösszeg-számító parancs használatára, például a *AIX Toolbox for Linux Applications CD* textutils RPM Package Manager csomagjában található **md5sum** parancsra.

Megbízható fájlok ellenőrzése:

A **tcbck** parancs segítségével ellenőrizze és javítsa ki a tcbck adatbázisban lévő fájlokat, és javítsa ki illetve állítsa elő az összes hiba naplóját.

A tcbck adatbázis összes fájljának ellenőrzéséhez, valamint az összes hiba kijavításához és kimutatásához írja be a következő parancsot:

```
tcbck -y ALL
```

Ennek hatására a **tcbck** parancs ellenőrzi a tcbck adatbázis összes fájljának telepítését, ahogy a /etc/security/sysck.cfg fájlban le van írva.

Ha mindezt automatikusan, a rendszer inicializálásakor kívánja végrehajtani és naplót kíván a hibákról, akkor a fenti parancssort írja hozzá a **/etc/rc** parancshoz.

A fájlrendszerfa ellenőrzése:

Ha bármikor arra gyanakszik, hogy a rendszer integritása sérült, akkor a fájlrendszer-fa ellenőrzéséhez futtassa le a **tcbck** parancsot.

A fájlrendszerfa ellenőrzéséhez írja be a következő parancsot:

```
tcbck -t tree
```

Ha a **tcbck** parancsot a *tree* értékkel együtt használja, akkor a rendszer összes fájljának telepítése ellenőrzésre kerül (ez hosszú ideig eltarthat). Ha a **tcbck** parancs talál olyan fájlt, amely esetleg veszélyeztetheti a rendszer biztonságát, a gyanús fájl módosítható és eltávolíthatók a rendet sértő attribútumok. Ezenfelül az alábbi ellenőrzések kerülnek elvégzésre a fájlrendszer összes többi fájlján:

- Ha a fájl tulajdonosa a root és a fájl SetUID bitje be van állítva, akkor a SetUID bit törlésre kerül.

- Ha a fájl csoportja adminisztrátori csoport, a fájl végrehajtható és a SetGID bit be van állítva, akkor a SetGID bit törlésre kerül.
- Ha a fájl **tcb** attribútuma be van állítva, akkor ez az attribútum törlésre kerül.
- Ha a fájl egy eszköz (karakter vagy blokk speciális fájl), akkor eltávolításra kerül.
- Ha a fájl egy további hivatkozás az /etc/security/sysck.cfg fájlban megadott elérési útra, akkor a hivatkozás törlődik.
- Ha a fájl egy további szimbolikus hivatkozás az /etc/security/sysck.cfg fájlban megadott elérési útra, akkor a szimbolikus hivatkozás eltávolításra kerül.

  Megjegyzés: A **tcbck** parancs végrehajtása előtt minden eszközbejegyzést hozzá kell adni a /etc/security/sysck.cfg fájlhoz, különben a rendszer használhatatlanná válhat. Megbízható eszközöket a /etc/security/sysck.cfg fájlhoz a **-l** kapcsoló segítségével lehet felvenni.

FIGYELEM: *NE* futtassa a **tcbck -y tree** parancsot. Ez a parancs töröl és letilt minden olyan eszközt, amelyik nincs tökéletesen megadva a TCB-ben, és ez használhatatlanná teheti a rendszert.

Megbízható program felvétele:

A **tcbck** parancs segítségével adjon hozzá egy adott programot a /etc/security/sysck.cfg fájlhoz.

Egy adott programnak az /etc/security/sysck.cfg fájlba történő felvételéhez írja be a következő parancsot:

```
tcbck -a útvonal [attribútum=érték]
```

Csak azokat az attribútumokat kell megadni a parancssorban, amelyek nem következnek a fájl aktuális állapotából. Az összes attribútumnév az /etc/security/sysck.cfg fájlban tárolódik.

Például az alábbi parancs bejegyzi a /usr/bin/setgroups nevű új SetUID root programot, amelyre a /usr/bin/getgroups hivatkozik:

```
tcbck -a /usr/bin/setgroups links=/usr/bin/getgroups
```

Ha a jfh és jsl elemet adminisztrátori felhasználóként, a developers elemet pedig adminisztrátori csoportként kívánja hozzáadni, hogy a /usr/bin/abc fájl biztonsági megfigyelése során ellenőrzésre kerüljön, akkor tegye a következőt:

```
tcbck -a /usr/bin/abc setuids=jfh,jsl setgids=developers
```

Egy program telepítése után előfordulhat, hogy nem tudja, mely új fájlok kerültek bejegyzésre az /etc/security/sysck.cfg fájlba. Ezek a fájlok az alábbi paranccsal kérhetők le és vehetők fel:

```
tcbck -t tree
```

Ez a parancs kiírja minden olyan fájl nevét, amelyet be kell jegyezni az /etc/security/sysck.cfg fájlban.

Megbízható program törlése:

Ha törölni kíván egy fájlt a rendszerből, amelynek leírása megtalálható az /etc/security/sysck.cfg fájlban, akkor az /etc/security/sysck.cfg fájlban található leírást is törölnie kell.

Ha például kitörölte az /etc/cvid programot, akkor az alábbi parancs kiadása hibaüzenetet eredményez:

```
tcbck -t ALL
```

Az eredményül kapott hibaüzenet a következő:

```
3001-020 A file /etc/cvid was not found.
```

A program leírása továbbra is megtalálható az /etc/security/sysck.cfg fájlban. A leírás törléséhez írja be az alábbi parancsot:

```
tcbck -d /etc/cvid
```

További megbízható beállítások megadása:

A Megbízható számítástechnikai alapkörnyezethez (TCB) további beállításokat is megadhat.

Terminál-hozzáférés korlátozása:

Az operációs rendszert beállíthatja a terminál-hozzáférés korlátozására.

A **getty** and **shell** parancs módosítja a terminál tulajdonosát és módját, így megakadályozza, hogy a nem megbízható programok elérjék a terminált. Az operációs rendszer lehetőséget ad a kizárólagos terminál-hozzáférés beállítására.

Biztonságos figyelem billentyű használata:

Megbízható kommunikációs elérési út a Biztonságos figyelem billentyű (SAK) fenntartott billentyűkombinációjának (Ctrl-X, majd Ctrl-R) lenyomásával hozható létre.

Megjegyzés: Körültekintően járjon el a SAK használatakor, mert ez leállítja az összes olyan folyamatot, amely a terminált próbálja elérni, illetve minden hivatkozást rá (a /dev/console például hivatkozhat a /dev/tty0-ra).

Megbízható kommunikációs elérési út az alábbi feltételek mellett hozható létre:

- A rendszerbe bejelentkezéskor

  A SAK lenyomása után:
  - Ha új bejelentkezési képernyő jelenik meg, akkor létrejött a biztonságos elérési út.
  - Ha a megbízható héj parancssor jelenik meg, akkor a kezdeti bejelentkezési képernyő egy jogosulatlan program volt, amelyik lehet, hogy megpróbálta ellopni a jelszavát. A **who** paranccsal állapítsa meg, hogy ki használja a terminált, majd jelentkezzen ki.
- Amikor azt szeretné, hogy a beírt parancs egy megbízható program futását eredményezze. Néhány példa erre:
  - Root felhasználóként futtatás. Csak azután futtasson bármit is root felhasználóként, ha létrehozott egy megbízható kommunikációs elérési utat. Ez garantálja, hogy egyetlen megbízhatatlan program sem futhat root felhasználói jogosultsággal.
  - A **su**, **passwd** és **newgrp** parancs futtatása. Ezeket a parancsokat csak egy megbízható kommunikációs elérési út létrehozása után futtassa.

Biztonságos figyelem billentyű beállítása:

Biztonságos figyelem billentyű beállítása megbízható kommunikációs elérési út létrehozásához.

Minden terminál külön konfigurálható, hogy a Biztonságos figyelem billentyű (SAK) lenyomása az adott terminálon megbízható kommunikációs elérési utat hozzon létre. Ezt az /etc/security/login.cfg fájl **sak_enabled** attribútuma adja meg. Ha az attribútum értéke Igaz, akkor a SAK be van kapcsolva.

Ha egy adott portot kommunikációra kíván használni (például az **uucp** paranccsal), akkor az /etc/security/login.cfg fájl a használt portra vonatkozó szakasza tartalmazza a következő sort:

```
sak_enabled = false
```

Ez a sor (vagy a szakasz hiányzó bejegyzése) letiltja a SAK működését az adott terminálon.

A SAK egy terminálon bekapcsolásához vegye fel a terminálhoz tartozó szakaszba az alábbi sort:

```
sak_enabled = true
```

Megbízható végrehajtás

A megbízható végrehajtás (TE) olyan szolgáltatások gyűjteményére hivatkozik, amelyek ellenőrzik a rendszer integritását valamint speciális biztonsági irányelvek valósítanak meg, amelyek által a teljes rendszer megbízhatósági szintje kiterjeszthető.

A rosszindulatú felhasználó általános rendszermegsértési módja a rendszerhozzáférés megszerzése, majd trójai állományok, rootkitek telepítése vagy néhány biztonság szempontjából kritikus fájl módosítása. Ezáltal a rendszer támadhatóvá és kihasználhatóvá válik. A megbízható végrehajtás szolgáltatáshalmaza mögött húzódó központi gondolat az ilyen tevékenységek megakadályozása vagy legrosszabb esetben ezek azonosítása, ha ilyen incidens történik a rendszeren. A megbízható végrehajtás által biztosított funkciók segítségével a rendszeradminisztrátor dönthet a futtatható végrehajtható fájlok tényleges halmazáról vagy a betölthető kernelbővítmények halmazáról. Ennek segítségével megfigyelhető a rendszer biztonsági állapota vagy azonosíthatók a módosított fájlok, ezáltal növelhető a rendszer megbízhatóságának szintje és megnehezíthető a rosszindulatú felhasználó számára a rendszer megsértése. A TE alatti szolgáltatások a következő csoportokba sorolhatók:

- Megbízható aláírás-adatbázis kezelése
- Megbízható aláírás-adatbázis integritásának megfigyelése
- Biztonsági irányelvek beállítása
- Megbízható végrehajtási útvonal és megbízható függvénytár-útvonal

Megjegyzés: Egy TCB funkcionalitás már létezik az AIX operációs rendszeren. A TE hatékonyabb és továbbfejlesztett mechanizmus, amely átfedi a TCB funkciók egy részét és továbbfejlesztett biztonsági irányelveket biztosít a rendszer integritásának jobb felügyelete érdekében. Mialatt a Megbízható számítástechnikai alapkörnyezet továbbra is rendelkezésre áll, a Megbízható végrehajtás bevezeti a rendszerintegritás ellenőrzésének és védelmének új és továbbfejlesztett alapelvét.

Megbízható aláírás-adatbázis kezelése:

A TCB-éhez hasonlóan létezik egy adatbázis, amely a rendszeren megtalálható megbízható fájlok kritikus biztonsági paramétereit tárolja. Ez a TSD adatbázis az /etc/security/tsd/tsd.dat fájlban található..

A *megbízható fájl* olyan fájl, amely a rendszer biztonsági szempontjából kritikus fontosságú, és veszélyeztetése esetén a teljes rendszer biztonsága veszélybe kerülhet. A leírásnak megfelelő fájlok jellemzően a következők:

- Kernel (operációs rendszer)
- Minden setuid root program
- Minden setgid root program
- A root felhasználó vagy a rendszercsoport egyik tagja által kizárólagosan futtatott programok
- Olyan programok, amelyeket az adminisztrátornak kell futtatnia a megbízható kommunikációs útvonalon (például az **ls** parancs)
- Rendszerműveletet vezérlő konfigurációs fájl
- A kernel vagy a rendszerkonfigurációs fájlok megváltoztatását lehetővé tevő jogosultságokkal futó program

Ideális esetben minden megbízható fájlnak rendelkeznie kell a TSD-ben tárolt társított szakasszal vagy fájlmeghatározással. A fájl megbízhatóként jelölhető a TSD-ben lévő meghatározásának felvételével a **trustchk** parancs segítségével. A **trustchk** parancs segítségével bejegyzések vehetők fel, törölhetők vagy listázhatók a TSD-ből.

Megbízható aláírás-adatbázis:

A megbízható aláírás-adatbázis a rendszeren található megbízható fájlok kritikus biztonsági paramétereinek tárolására használt adatbázis. Ez az adatbázis az /etc/security/tsd/tsd.dat könyvtárban található.

Ideális esetben minden megbízható fájlnak rendelkeznie kell a TSD-ben tárolt társított szakasszal vagy fájlmeghatározással. Minden megbízható fájlhoz egyedi kriptográfiai kivonat és digitális aláírás kerül hozzárendelésre. A megbízható fájlok alapértelmezett készletének kriptográfiai kivonatát az SHA-256 algoritmus segítségével állítja elő az AIX összeállítási környezet, és az AIX telepítési fájlkészletek részeként kerül csomagolásra. Ezek a kivonatértékek és az aláírások a megfelelő AIX telepítési képfájlok részét képezik és a célrendszeren a megbízható szoftveradatbázis (/etc/security/tsd/tsd.dat) tárolja, a következő példaszakasz formátumában:

```
/usr/bin/ps:
          owner          = bin
          group          = system
```

```
           mode           = 555
           type           = FILE
           hardlinks      = /usr/sbin/ps
           symlinks       =
           size           = 1024
           cert_tag       = bbe21b795c550ab243
           signature      =
f7167eb9ba3b63478793c635fc991c7e9663365b2c238411d24c2a8a
           hash_value     = c550ab2436792256b4846a8d0dc448fc45
           minslabel      = SLSL
           maxslabel      = SLSL
           intlabel       = SHTL
           accessauths    = aix.mls.pdir, aix.mls.config
           innateprivs    = PV_LEF
           proxyprivs     = PV_DAC
           authprivs      =
aix.security.cmds:PV_DAC,aix.ras.audit:PV_AU_ADMIN
           secflags       = FSF_EPS
           t_accessauths  =
           t_innateprivs  =
           t_proxyprivs   =
           t_authprivs    =
           t_secflags     =
```

owner A fájl tulajdonosa. Ezt az értéket a **trustchk** parancs számítja ki a fájl TSD-hez adásakor.

group A fájl csoportja. Az értéket a **trustchk** parancs számítja ki.

mode Értékek vesszővel elválasztott listája. Megengedett értékek: **SUID** (SUID set bit), **SGID** (SGID set bit), **SVTX** (SVTX set bit) és **TCB** (Megbízható számítástechnikai alapkörnyezet). A fájljogosultságnak az utolsó értéknek kell lennie, és oktális értékként adható meg. Az **rwxr-xr-x** jogosultságbitekkel rendelkező **uid** beállítású fájlnak például a módértéke **SUID,755**. Az értéket a **trustchk** parancs számítja ki.

type A fájl típusa. Az értéket a **trustchk** parancs számítja ki. Lehetséges értékek: **FILE**, **DIRECTORY**, **MPX_DEV**, **CHAR_DEV**, **BLK_DEV** és **FIFO**.

hardlinks
A fájl közvetlen hivatkozásainak listája. Az értéket a **trustchk** parancs nem tudja kiszámítani. Ezt a felhasználó adja meg fájl adatbázishoz adásakor.

symlinks
A fájl szimbolikus hivatkozásainak listája. Az értéket a **trustchk** parancs nem tudja kiszámítani. Ezt a felhasználó adja meg a fájl adatbázishoz adásakor.

size Megadja a fájl méretét. A **VOLATILE** érték azt jelenti, hogy a fájl gyakran módosításra kerül.

cert_tag
Ez a mező leképezi a fájl digitális aláírását a társított tanúsítvánnyal, amelynek használatával ellenőrizhető a fájl aláírása. Ez a mező tárolja a tanúsítványazonosítót, és a **trustchk** parancs számítja ki a fájl TSD-hez adásakor. A tanúsítványokat az /etc/security/certificates könyvtár tárolja.

signature
A fájl digitális aláírása. A **VOLATILE** érték azt jelenti, hogy a fájl gyakran módosításra kerül. A mezőt a **trustchk** parancs számítja ki.

hash_value
A fájl kriptográfiai kivonata. A **VOLATILE** érték azt jelenti, hogy a fájl gyakran módosításra kerül. A mezőt a **trustchk** parancs számítja ki.

minslabel
Az objektum minimális érzékenységi címkéjét adja meg.

maxslabel
Az objektum maximális érzékenységi címkéjét adja meg (Megbízható AIX rendszeren érvényes). Ez az attribútum szabályos fájlokra és fifo-ra nem alkalmazható.

intlabel
Az objektum integritási címkéjét adja meg (Megbízható AIX rendszeren érvényes).

accessauths
Az objektum hozzáférési jogosultságát adja meg (Megbízható AIX rendszeren érvényes).

innateprivs
A fájl belső jogosultságait adja meg.

proxyprivs
A fájl proxyjogosultságait adja meg.

authprivs
A felhasználóhoz az adott felhatalmazások után hozzárendelt jogosultságokat adja meg.

secflags
Az objektumhoz rendelt fájlbiztonsági kapcsolókat adja meg.

t_accessauth
További Trusted AIX rendszert ad meg többszintű biztonságra (MLS) jellemző hozzáférési jogosultsággal (Megbízható AIX rendszeren érvényes).

t_innateprivs
További Trusted AIX rendszert ad meg a fájlhoz MLS-re jellemző belső jogosultságokkal (Trusted AIX rendszeren érvényes).

t_proxyprivs
További Trusted AIX rendszert ad meg a fájlhoz MLS-re jellemző proxy jogosultságokkal (Trusted AIX rendszeren érvényes).

t_authprivs
További Trusted AIX rendszert ad meg MLS-re jellemző jogosultságokkal, amelyek a felhasználóhoz az adott felhatalmazások után kerülnek hozzárendelésre (Trusted AIX rendszeren érvényes).

t_secflags
További Trusted AIX rendszert ad meg az objektumhoz rendelt MLS-re jellemző fájlbiztonsági kapcsolókkal (Trusted AIX rendszeren érvényes).

Új bejegyzés TSD-hez adásakor ha a megbízható fájl rendelkezik arra mutató szimbolikus vagy közvetlen hivatkozással, akkor ezek a hivatkozások a **symlinks** és **hardlinks** attribútummal hozzáadhatók a TSD-hez a parancssorban, a **trustchk** paranccsal együtt. Ha a hozzáadni kívánt fájl várhatóan gyakran változik, akkor a parancssorban használja a VOLATILE kulcsszót. A **trustchk** parancs nem számítja ki a **hash_value** és **signature** mezőt, amikor előállítja a fájlmeghatározást a TSD-hez adás érdekében. A fájl integritásellenőrzése során a **hash_value** és **signature** mező figyelmen kívül marad.

Szokásos fájlmeghatározások TSD-hez adása során meg kell adni egy magánkulcsot (ASN.1/DER formátum). Használja a **-s** paramétert és a digitális tanúsítványt a megfelelő nyilvános kulccsal a **-v** paraméter alkalmazásával. A magánkulcs előállítja a fájl aláírását, majd eldobásra kerül. A felhasználó felelőssége a kulcs biztonságos tárolása. A tanúsítvány az /etc/security/certificates fájl tanúsítványtárolójában kerül tárolásra annak érdekében, hogy az aláírásokat integritásellenőrzés kérésekor ellenőrizni lehessen. Mivel az aláírás-kiszámítás nem szokásos fájlok - mint például a könyvtár és az eszközfájl - esetén nem lehetséges, ezek TSD-hez adása esetén meg kell adni a magánkulcsot és a tanúsítványt.

Az előre kiszámított meghatározás a TSD-hez adni kívánt fájlon keresztül is biztosítható az **-f** paraméter segítségével. Ebben az esetben a **trustchk** parancs nem számítja ki az értékeket és ellenőrzés nélkül tárolja a meghatározásokat a TSD-ben. Ebben az esetben a felhasználó felelős a fájlmeghatározások értelméért.

Függvénytár ellenőrzésének támogatása

A függvénytár-ellenőrzés támogatása érdekében felvételre kerül a tsd.dat fájl az /etc/security/tsd/lib/ könyvtárba. Az adatbázis neve: /etc/security/tsd/lib/lib.tsd.dat. Ez az adatbázis kifejezetten azoknak a függvénytáraknak van, amelyek a vonatkozó megbízható könyvtár .o fájljainak szakaszait tartalmazzák. Egy függvénytár minden .o fájljához tartozó szakasz az alábbi példában megadott formátumú.

A libc.a függvénytár esetén ha az strcmp.o az .o fájltípus egyike, a strcmp.o fájlhoz tartozó szakasz az /etc/security/tsd/lib/lib.tsd.dat fájlban az alábbi példához hasonló:

```
/usr/lib/libc.a/strcmp.o:
   Type = OBJ
   Size = 2345
   Hash value
   Signature =
   Cert_tag =
```

Ez az adatbázis az .o fájlhoz tartozó **type**, **size hash**, **cert tag** és **signature** bejegyzéseket tartalmaz. A függvénytár kivonata frissül az /etc/security/tsd/tsd.dat fájlban a megfelelő szakasznál. Ezek az attribútumértékek dinamikusan kerülnek előállításra az összeállítás során, és az értékek áthelyezésre kerülnek az /etc/security/tsd/lib/lib.tsd.dat adatbázisba a telepítés során.

Az /etc/security/tsd/tsd.dat fájlban módosulnak a függvénytárakhoz tartozó szakaszok, hogy a **type** attribútum LIB legyen, a **size** és a **signature** attribútum pedig üres. Jelenleg a **dynamica** attribútumok (**size**, **hash**, **signature**) értékeit **VOLATILE** értékekként kezeli a rendszer. Ezért a függvénytár-ellenőrzés kimarad a rendszerbetöltés során. Az AIX 6.1.0 kiadásával kezdődően a megbízható függvénytár szakaszok **size**, **hash** és **signature** értékét a függvénytárhoz tartozó .o fájlokkal számítja ki a rendszer. A telepítés során a tsd.dat adatbázis feltöltésre kerül a kiszámított értékekkel, a megbízható függvénytárhoz tartozó megfelelő .o fájl szakasz pedig az /etc/security/tsd/lib/lib.tsd.dat adatbázisban kerül tárolásra.

Távoli TE adatbázis hozzáférés:

Központosított megbízható aláírási adatbázis (TSD) irányelvek és megbízható végrehajtási (TE) irányelvek a rendszerkörnyezetében úgy valósíthatók meg, ha LDAP címtárban tárolja őket.

A TSD és TE irányelveket vezérlő adatbázis tárolása az egyes rendszerektől független. AIX A központosított TSD irányelvek és TE irányelvek tárolása LDAP címtárban történik, hogy azokat központilag lehessen kezelni. A központosított TSD irányelvek és TE irányelvek használata lehetővé teszi annak ellenőrzését, hogy az irányelvek az LDAP címtárban a mesterpéldányok és az irányelvek a kliens újratelepítésekor, frissítésekor vagy a biztonság megsértésekor frissíthetik a klienseket. A központosított TE irányelvek lehetővé teszik a TE irányelvek egy helyről történő betartatását anélkül, hogy minden klienst külön frissíteni kellene. A központosított TSD irányelveket sokkal könnyebb kezelni, mint a nem központosított TSD irányelveket.

Az AIX Segédprogramok használatával a helyi TSD és TE irányelvek adatai exportálhatók LDAP címtárba, konfigurálhatók a kliensek az LDAP címtárban lévő TSD és TE irányelvek használatára, vezérelhető a TSD és TE irányelvek adatainak kikeresése, és az LDAP adatok a kliensrendszerről kezelhetők. Az alábbi részek további információkat nyújtanak ezekről a szolgáltatásokról.

TSD irányelvek és TE irányelvek adatainak exportálása LDAP címtárba:

Ahhoz, hogy a TSD és TE irányelvek központi lerakataként az LDAP címtárt használja, az LDAP szervert fel kell tölteni az irányelvek adataival.

Az LDAP szerverre telepíteni kell a TSD és TE irányelvek sémáit, hogy az LDAP kliensek az irányelvek adataihoz használhassák a szervert. Az LDAP TSD és TE irányelveinek sémája az AIX rendszer **/etc/security/ldap/sec.ldif** fájljában érhető el. Az LDAP szerver sémáját frissíteni kell ezzel a fájllal az **ldapmodify** parancs segítségével.

Hogy azonosítsa az LDAP szerver TE adatbázisainak változatát, és hogy az LDAP kliensek tudatában legyenek az adott változatnak, be kell állítania a **databasename** attribútumot a **/etc/nscontrol.conf** fájlban. A **databasename** attribútum bármilyen nevet elfogad értékként, és a **tetoldif** parancs használja az ldif formátum létrehozásakor.

A **tetoldif** parancs segítségével olvassa be a helyi TSD és TE irányelvfájlok adatait és készítsen az LDAP által használható formátumú kimenetet. A **tetoldif** parancs által készített kimenet elmenthető ldif formátumú fájlba, majd az **ldapadd** parancs segítségével felhasználható az LDAP szerver feltöltésére az adatokkal. A helyi rendszer következő adatbázisait használja a **tetoldif** parancs a TSD és TE irányelv adatok előállításához az LDAP számára:

- /etc/security/tsd/tsd.dat
- /etc/security/tsd/tepolicies.dat

LDAP klienskonfiguráció TSD és TE irányelvekhez:

A rendszert LDAP kliensként kell beállítani az LDAP-ban tárolt TSD és TE irányelvadatok használatához.

Az AIX **/usr/sbin/mksecldap** parancs segítségével beállíthatja a rendszert LDAP kliensként. Az **mksecldap** parancs dinamikus keresést végez a megadott LDAP szerveren a TSD és TE irányelvadatok helyének meghatározásához, és az eredményeket a **/etc/security/ldap/ldap.cfg** fájlba menti.

Miután a **mksecldap** paranccsal sikeresen konfigurálta a rendszert LDAP kliensként, a **/etc/nscontrol.conf** fájlban a secorder konfigurálásával engedélyezni kell, hogy az LDAP legyen a TSD és TE irányelvadatok kikeresési tartománya.

Miután a rendszert konfigurálta LDAP kliensként és TSD és TE irányelvadatok kikeresési tartományaként, a **/usr/sbin/secldapclntd** kliensdémon beolvassa a TSD és TE irányelvadatokat az LDAP szerverről, ha **trustchk** parancsokat hajtanak végre az LDAP kliensen.

LDAP engedélyezése a trustchk paranccsal:

Minden TSD irányelv és TE irányelv adatbázis-kezelő parancsa engedélyezve van az LDAP TSD és TE irányelvek adatbázisának használatára.

A **trustchk** parancsot az **–R** kapcsolóval használva hajtsa végre az LDAP adatbázis kezdeti beállítását. A kezdeti beállítás tartalmazza TSD irányelvek, TE irányelvek, kiindulási DN-ek hozzáadását és a helyi **/etc/security/tsd/ldap/tsd.dat** és **/etc/security/tsd/ldap/tepolicies.dat** adatbázisfájlok létrehozását.

Ha a **trustchk** parancsot az **–R** kapcsolóval és az LDAP beállítással futtatja, akkor a műveleteknek az LDAP szerver adatai képezik az alapját. Ha a **trustchk** parancsot az **–R** kapcsolóval és a files beállítással futtatja, akkor a műveleteknek a helyi adatbázis adatai képezik az alapját. Az **–R** kapcsoló alapértelmezése a files beállítás használata.

Kapcsolódó tájékoztatás:

mksecldap parancs

trustchk parancs

Megbízható aláírás-adatbázis integritásának megfigyelése:

A **trustchk** parancs segítségével megfigyelhető a Megbízható aláírás-adatbázisban (TSD) lévő fájlmeghatározások integritási állapota az aktuális fájlokkal szemben.

Ha a **trustchk** parancs anomáliát jelez, akkor az automatikusan kijavítható vagy a javítás előtt a felhasználó megkérdezhető. A méret, alírás, cert_tag vagy hash_value eltéréshez hasonló anomáliák esetén a javítás nem lehetséges. Ilyen esetekben a **trustchk** parancs hatására a fájl elérhetetlenné válik, ezáltal a feldolgozása értelmetlen és a fájl sérült.

A következő javítótevékenységek kerülnek végrehajtásra a különböző eltérő attribútumokhoz:

tulajdonos
A fájl tulajdonosa visszaállításra kerül a TSD-ben lévő értékre.

csoport
A csoport tulajdonosa visszaállításra kerül a TSD-ben lévő értékre.

mód A fájl mód bitjei visszaállításra kerülnek a TSD-ben lévő értékre.

hardlinks
Ha a hivatkozás másik fájlra mutat, akkor módosításra kerül, hogy az adott fájlra mutasson. Ha a hivatkozás nem létezik, akkor egy új hivatkozás kerül létrehozásra, amely az adott fájlra mutat.

symlinks
Ugyanaz, mint a közvetlen hivatkozások.

type A fájl nem elérhető.

size A fájl nem elérhető, a **VOLATILE** fájl kivételével.

cert_tag
A fájl nem elérhető.

signature
A fájl nem elérhető, a **VOLATILE** fájl kivételével.

hash_value
A fájl nem elérhető, a **VOLATILE** fájl kivételével.

minslabel
Megbízható AIX rendszeren a minimális érzékenységi szint visszaállításra kerül a TSD értékére.

maxslabel
Megbízható AIX rendszeren a maximális érzékenységi szint visszaállításra kerül a TSD értékére.

intlabel
Megbízható AIX rendszeren az integritási szint visszaállításra kerül a TSD értékére.

accessauths
A hozzáférési jogosultságok visszaállításra kerülnek a TSD értékére. Megbízható AIX rendszeren a **t_accessauths** értékek az **accessauths** attribútum részei.

innateprivs
A belső jogosultságok visszaállításra kerülnek a TSD értékére. Megbízható AIX rendszeren a **t_innateprivs** értékek az **innateprivs** attribútum részei.

inheritprivs
Az örökölhető jogosultságok visszaállításra kerülnek a TSD értékére. Megbízható AIX rendszeren a **t_inheritprivs** értékek az öröklés attribútum részei.

authprivs
A felhatalmazott jogosultságok visszaállításra kerülnek a TSD értékére. Megbízható AIX rendszeren a **t_authprivs** értékek az **authprivs** attribútum részei.

aecflags
A biztonsági kapcsolók visszaállításra kerülnek a TSD értékére. Megbízható AIX rendszeren a **t_secglags** értékek a **secflags** attribútum részei.

A **-F** paraméterrel ellenőrizhetők egy alternatív adatbázisra fájlmeghatározásai. A rendszeradminisztrátornak el kell kerülnie a TSD tárolását ugyanazon a rendszeren és az adatbázis mentését alternatív helyre. A **-F** paraméterrel ez a fájlintegritás megfeleltethető a TSD mentett változatának.

Biztonsági irányelvek beállítása:

A megbízható végrehajtás (TE) szolgáltatás futás közbeni fájlintegritás-ellenőrzési mechanizmust biztosít. A mechanizmus segítségével a rendszer beállítható a megbízható fájlok integritásának ellenőrzésére, mielőtt minden kérés elérné ezeket a fájlokat, és csak az integritásellenőrzés által elfogadott megbízható fájlok elérését teszi lehetővé a rendszeren.

Ha a fájl megbízhatóként van jelölve (a meghatározásának Megbízható aláírás-adatbázishoz adásával), akkor a TE szolgáltatás segítségével minden hozzáférés esetén megfigyelhető az integritása. A TE folyamatosan meg tudja figyelni a rendszert és felismeri a rendszeren lévő megbízható fájl (rosszindulató felhasználó vagy alkalmazás általi) módosítását futási időben (például betöltéskor). Ha a rendszer úgy találta, hogy a fájl módosítva lett, akkor TE javító tevékenységeket végezhet az előre beállított irányelvek alapján, mint például a végrehajtás tiltása, fájlelérés vagy hibanaplózás. Ha egy fájl megnyitásra vagy végrehajtásra kerül, és bejegyzéssel rendelkezik a Megbízható aláírás-adatbázisban (TSD), akkor a TE a következőt teszi:

- A bináris fájl betöltése előtt a fájl betöltéséért felelős összetevő (rendszerbetöltő) meghívja a Megbízható végrehajtási alrendszert, és kiszámítja a kivonatértéket az SHA-256 algoritmus (beállítható) segítségével.
- Ez a futás közben kiszámított kivonatérték összehasonlításra kerül a TSD-ben tárolttal.
- Ha az értékek egyeznek, akkor a fájlmegnyitás vagy végrehajtás engedélyezett.
- Ha az értékek nem egyeznek meg, akkor a bináris fájl módosítva, vagy egyéb módon veszélyeztetve lett. A felhasználónak kell dönteni a végrehajtandó tevékenységről. A TE mechanizmus lehetőséget biztosít a felhasználók számára saját irányelvek beállítására a kivonatértékek eltérése esetén végrehajtandó tevékenységekhez.
- A beállított irányelvek alapján végrehajtásra kerül egy megfelelő tevékenység.

A következő irányelvek állíthatók be:

CHKEXEC
: Csak a megbízható végrehajtható fájlok kivonatértékét ellenőrzi, mielőtt betöltené őket a memóriába végrehajtásra.

CHKSHLIBS
: Csak a megbízható osztott könyvtárak kivonatértékét ellenőrzi, mielőtt betöltené őket a memóriába végrehajtásra.

CHKSCRIPTS
: Csak a megbízható parancsértelmező-parancsfájlok kivonatértékét ellenőrzi, mielőtt betöltené őket a memóriába.

CHKKERNEXT
: Csak a kernelbővítmény kivonatértékét ellenőrzi a memóriába való betöltés előtt.

STOP_UNTRUSTD
: Leállítja a nem megbízható fájlok betöltését. Csak a TSD-hez tartozó fájlok kerülnek betöltésre. Ez az irányelv csak a fent említett CHK* irányelvek kombinációjával működik. Ha például a **CHKEXEC=ON** és **STOP_UNTRUSTD=ON** van megadva, akkor a nem a TSD-hez tartozó végrehajtható bináris fájlok nem kerülnek végrehajtásra.

STOP_ON_CHKFAIL
: Leállítja azon megbízható fájlok betöltését, amelyek nem mennek át a kivonatérték-ellenőrzésen. Ez az irányelv a CHK* irányelvekkel együtt is működik. Ha például a **CHKSHLIBS=ON** és **STOP_ON_CHKFAIL=ON** meg van adva, akkor a TSD-hez nem tartozó osztott függvénytár nem kerül betöltésre a memóriába felhasználásra.

TSD_LOCK
: Zárolja a TSD-t, ezért az nem szerkeszthető.

TSD_FILES_LOCK
: Zárolja a megbízható fájlokat. Ez nem engedélyezi a megbízható fájlok megnyitását írás módban.

TE Megbízható végrehajtási funkció engedélyezése/tiltása. A fent említett irányelvek csak akkor vannak hatályban, ha ez engedélyezett.

A következő tábla a különböző CHK* és STOP* irányelvek közötti interakciót adja meg, ha engedélyezett:

Irányelv	STOP_UNTRUSTD	STOP_ON_CHKFAIL
CHKEXEC	Leállítja a nem a TSD-hez tartozó végrehajtható fájlok betöltését.	Leállítja azon végrehajtható fájlok betöltését, amelyek kivonatértéke nem felel meg a TSD értékeknek.
CHKSHLIBS	Leállítja a nem a TSD-hez tartozó osztott függvénytárak betöltését.	Leállítja azon osztott függvénytárak betöltését, amelyek kivonatértéke nem felel meg a TSD értékeknek.
CHKSCRIPTS	Leállítja a nem a TSD-hez tartozó parancsértelmező-parancsfájlok betöltését.	Leállítja azon parancsértelmező-parancsfájlok betöltését, amelyek kivonatértéke nem felel meg a TSD értékeknek.
CHKKERNEXT	Leállítja a nem a TSD-hez tartozó kernelbővítmények betöltését.	Leállítja azon kernelbővítmények betöltését, amelyek kivonatértéke nem felel meg a TSD értékeknek.

Megjegyzés: Az irányelv bármikor engedélyezhető vagy letiltható, amíg a TE nincs bekapcsolva az irányelvek hatályba léptetéséhez. Ha egy irányelv hatályban van, akkor az irányelv letiltása csak a következő rendszerbetöltési ciklusban lép hatályba. Az információs üzenetek a **syslog** naplóban kerülnek naplózásra.

Kapcsolódó tájékoztatás:

TE_verify_reg kernelszolgáltatás

TE_verify_unreg kernelszolgáltatás

Megbízható végrehajtási útvonal és megbízható függvénytár-útvonal:

A megbízható végrehajtási útvonal (TEP) a megbízható végrehajtható fájlokat tartalmazó könyvtárak listáját adja meg. A TEP ellenőrzés engedélyezése után a rendszerbetöltő lehetővé teszi a megadott útvonalakon lévő bináris fájlok végrehajtását. A megbízható függvénytár-útvonal (TLP) működése ezzel egyező, azzal a különbséggel, hogy a rendszer megbízható függvénytárait tartalmazó könyvtárakat adja meg.

A TLP engedélyezése után a rendszerbetöltő csak ezen az útvonalon lévő függvénytárak bináris fájlokhoz való kötését teszi lehetővé. A **trustchk** parancs segítségével engedélyezhető vagy letiltható a TEP vagy TLP, valamint beállítható mindkettő kettősponttal elválasztott útvonallistája, a **trustchk** parancs TEP és TLP parancssori attribútumai segítségével.

Megbízható parancsértelmező és Attn billentyű:

A megbízható parancsértelmező és Attn billentyű (SAK) a Megbízható számítástechnikai alapkörnyezethez (TCB) hasonlóan működik azzal a kivétellel, hogy ha TCB helyett a megbízható végrehajtás engedélyezett a rendszeren, akkor a megbízható parancsértelmező csak a megbízható aláírás-adatbázishoz tartozó fájlokat hajtja végre.

A TCB-vel és SAK-val kapcsolatos további információkat a Megbízható számítástechnikai alapkörnyezet, a Biztonságos figyelem billentyű használata és a Biztonságos figyelem kulcs beállítása rész tartalmaz.

Megbízható végrehajtási (TE) irányelvek adatbázisa:

A megbízható végrehajtási (TE) irányelvek a **/etc/security/tsd/tepolicies.dat** fájlban találhatók. A TE irányelvek útvonala a TLP és TEP könyvtárakkal van felsorolva.

Biztonsági profil Kiértékelés biztosítási szint 4+ and Címkézett AIX Biztonság és Kiértékelés biztosítási szint 4+

A rendszeradminisztrátor a rendszert Base AIX Security (BAS) és Evaluation Assurance Level 4+ (EAL4+) beállítással vagy Labeled AIX Security (LAS) és Kiértékelés biztosítási szint 4+ (EAL4+) lehetőséggel telepítheti az alap operációs rendszer (BOS) telepítés során. Ezekkel a beállításokkal telepített rendszer korlátozza a BOS telepítés során telepített szoftvereket, valamint a hálózati hozzáférést.

Megjegyzés: Az AIX Version 7.1 kiértékelése jelenleg fut. Tekintse meg az AIX Version 7.1 kiadási megjegyzéseit a legújabb információkért.

Biztonsági profil áttekintése:

A Biztonsági profil egy olyan termék, amely biztonsági követelményeket határoz meg a hálózati környezetekben található általános célú operációs rendszerekhez. A profil meghatározza a Kiértékelési cél (TOE) biztonsági funkcióhoz és környezetéhez tartozó biztonsági célok eléréséhez szükséges követelményeket.

A Biztonsági profil egy alap csomagot és számos kiegészítő csomagot tartalmaz. A Biztonsági profil alap csomag támogatásához kapcsolódó termékek az Azonosítás és hitelesítés, a Tetszés szerinti hozzáférés-felügyelet (DAC), a Felülvizsgálat, a Kriptográfiai szolgáltatások, a Felügyeleti és biztonsági mechanizmusok, valamint a Megbízható csatorna kommunikáció. A Biztonsági profil további, nem kötelező csomagokat is tartalmaz a Címkézett biztonság, az Integritás ellenőrzése, a Speciális felülvizsgálat, az Általános célú kriptográfia, a Speciális felügyelet, a Kiterjesztett azonosítás és hitelesítés, a Megbízható rendszerbetöltés, valamint a Virtualizáció termékhez.

Feltételezések

- A TOE funkcióhoz használható környezet:

  A szakaszban szereplő feltételezések mindegyike Base AIX Security (BAS üzemmód) és Labeled AIX Security (LAS üzemmód) környezetekre vonatkozik, kivéve, ha másképp van jelezve. A Virtuális I/O szerverre (VIOS) vonatkozó feltételezések kifejezetten meg vannak jelölve, hogy csak VIOS szerverre érvényesek. A VIOS nem osztozik a feltételezéseken sem az AIX, sem a Trusted AIX operációs rendszerrel.
- Fizikai:

  Az IT környezet megfelelő fizikai biztonsági szolgáltatásokkal nyújtja a TOE-t, amelyek arányban állnak a TOE által védett IT eszközök értékével.

  Megjegyzés: Csak VIOS esetén: A működési környezet megfelelő fizikai biztonsági szolgáltatásokkal nyújtja a TOE-t, amelyek arányban állnak a TOE által védett IT eszközök értékével.
- Adminisztráció:
  - A TOE biztonsági funkciót egy vagy több kompetens személy kezeli. A rendszeradminisztrációs személyzet nem gondatlan, szándékosan hanyag vagy rosszindulatú, és ragaszkodik a segítségnyújtó dokumentáció által biztosított utasításokhoz.
  - A jogosult felhasználók hozzáférhetnek a TOE által kezelt bizonyos információkhoz, és elvért tőlük az együttműködő hozzáállás.
  - A felhasználók megfelelően be vannak tanítva és megbízhatóak, hogy elvégezzenek bizonyos feladatokat vagy feladatcsoportokat a védett IT környezeten belül. Teljes felügyelettel rendelkeznek a saját felhasználói adataikat illetően.
  - Csak VIOS esetén: A TOE biztonsági funkciót egy vagy több kompetens személy kezeli. A rendszeradminisztrációs személyzet nem gondatlan, szándékosan hanyag vagy rosszindulatú, és ragaszkodik a segítségnyújtó dokumentáció által biztosított utasításokhoz.
  - Csak VIOS esetén: A jogosult felhasználók rendelkeznek a szükséges felhatalmazással a TOE által kezelt információknak legalább egy részéhez, és elvért tőlük az együttműködő hozzáállás.
  - Csak VIOS esetén: A felhasználók megfelelően be vannak tanítva és megbízhatóak, hogy elvégezzenek bizonyos feladatokat vagy feladatcsoportokat a védett működési környezeten belül. Teljes felügyelettel rendelkeznek a saját felhasználói adataikat illetően.
- Ügyrendi:
  - Az adminisztrátori felhasználónak észlelnie kell a TOE biztonságot betartató vagy biztonsággal kapcsolatos fájljainak bármilyen módosítását vagy sérülését, amelyet a felhasználó vagy az alapul szolgáló platform okozott szándékosan vagy véletlenül.
  - Minden távoli megbízható IT rendszerről, amelyben megbízik a Cél biztonsági funkció (TSF), hogy TSF adatokat vagy szolgáltatásokat biztosítson a TOE számára, vagy támogassa a TSF-et a biztonsági irányelv döntések betartatásában, feltételezzük, hogy ugyanazon kezelési vezérlés alatt van és olyan biztonsági irányelv megszorítások alatt működik, ami kompatibilis a TOE biztonsági irányelvével.

– Minden távoli megbízható IT rendszerről, amelyben megbízik a TSF, hogy TSF adatokat vagy szolgáltatásokat biztosítson a TOE számára, vagy támogassa a TSF-et a biztonsági irányelv döntések betartatásában, feltételezzük, hogy megfelelően megvalósítja a TSF által használt funkciókat és konzisztens a funkcióhoz meghatározott feltételezésekkel.
– Az alábbi információk integritása biztosított:
  - Minden TSF kód, beleértve az integritásellenőrzési funkciót is, amelyet az integritásellenőrzési mechanizmus indítása előtt betölt és futtat a rendszer.
  - Minden TSF adat, beleértve az integritást ellenőrző TSF adatokat is, amelyeket az integritásellenőrzési mechanizmus indítása előtt betöltött és futtatott TSF kód használ.
– Csak VIOS esetén: Az adminisztrátori felhasználónak észlelnie kell a TOE biztonságot betartató vagy biztonsággal kapcsolatos fájljainak bármilyen módosítását vagy sérülését, amelyet a felhasználó vagy az alapul szolgáló platform okozott szándékosan vagy véletlenül.

- Összekapcsolhatóság: A távoli megbízható IT rendszerekkel és rendszerektől, valamint a TSF fizikailag különálló, nem maga a TSF által védett részei között kialakított összes kapcsolat fizikailag vagy logikailag védett a TOE környezeten belül az átvitt adatok integritásának és bizalmasságának, valamint a kommunikációs végpontok eredetiségének biztosítása érdekében.

Szoftver beszerzése

A szoftver beszerzéséhez tegye a következőket:

1. Töltse le a terméket.
2. Kattintson a Súgó menüpontra a Jogosult szoftver támogatás menüben a bal oldali panelen. Az Általános feltételek kiértékelt konfiguráció megköveteli, hogy fizikai adathordozón vagy a letöltésirányító használatával szerezze be a terméket és az esetleges frissítéseket.

A termék telepítésével kapcsolatos információk: BAS/EAL4+ rendszer telepítése.

BAS /EAL4+ rendszer telepítése:

Az RBAC automatikusan engedélyezett, amikor ez a lehetőség ki van választva.

Ha be akarja állítani a BAS/EAL4+ lehetőséget egy BOS telepítés során, akkor tegye a következőket:

1. A Telepítés és beállítások képernyőn válassza a **További lehetőséget** elemet.
2. A További lehetőségek alatt állítson be **Igen** értéket a BAS/EAL4+ elemnél, és LPAR használata esetén állítson be **Nem** értéket a TCB elemnél. Ha egyéni bosinst.data fájlt használ csendes telepítéshez, akkor a TCB elem állítható **Igen** értékűre is.

Tiltsa le a távoli root bejelentkezést a BAS telepítésbe. A távoli root bejelentkezés letiltásához futtassa az alábbi parancsot a telepítés után:

```
/usr/bin/chuser rlogin=false subgroups=SUADMIN root
```

Vegye fel az adminisztrátori felhasználókat a **SUADMIN** csoportba, hogy lehetőségük legyen **su** paranccsal átváltani a root felhasználóra.

A **BAS és EAL4+ technológia engedélyezése** lehetőség csak az alábbi feltételek teljesülése esetén érhető el:

- A telepítési módszer új és teljes felülírás telepítésre van állítva.
- Az Angol nyelv van kiválasztva.
- A 64 bites kernel engedélyezett.
- A kiterjesztett naplózott fájlrendszer (JFS2) engedélyezett.

Amikor a **BAS és EAL4+ technológia engedélyezése** beállítás értéke Igen, akkor a **Megbízható számítástechnikai alapkörnyezet** beállítás is Igen, és az érvényes **Asztal** választások: NONE vagy CDE.

Ha csendes telepítést végez egyéni bosinst.data fájl használatával, akkor az **INSTALL_TYPE** mezőt CC_EVAL értékre kell állítani, a következő mezőket pedig az alábbiak szerint kell beállítani:

```
control_flow:
 CONSOLE = ???
 PROMPT = yes
 INSTALL_TYPE = CC_EVAL
 INSTALL_METHOD = overwrite
 TCB = yes
 DESKTOP = NONE vagy CDE
 ENABLE_64BIT_KERNEL = yes
 CREATE_JFS2_FS = yes
 ALL_DEVICES_KERNELS = no
 FIREFOX_BUNDLE = no
 HTTP_SERVER_BUNDLE = no
 KERBEROS_5_BUNDLE = no
 SERVER_BUNDLE = no
 ALT_DISK_INSTALL_BUNDLE = no

locale:
 CULTURAL_CONVENTION = en_US vagy C
 MESSAGES = en_US vagy C
```

További RBAC információkat itt talál: Szerep alapú hozzáférés-felügyelet (RBAC).

Hálózati telepítéskezelés környezet BAS/EAL4+ esetén:

A BAS/EAL4+ technológia ügyfelek telepítése végrehajtható a Hálózati telepítéskezelés (NIM) környezet használatával.

A NIM vezérlő úgy van beállítva, hogy biztosítsa a szükséges erőforrásokat az AIX 7.1 megfelelő BAS/EAL4+ szintjének telepítéséhez. Ezután a NIM ügyfelek a NIM vezérlőn található erőforrások használatával telepíthetők. Végrehajthatja az ügyfél csendes NIM telepítését az alábbi mezők beállításával a **bosinst_data** erőforrásban:

```
control_flow:
 CONSOLE = ???
 PROMPT = no
 INSTALL_TYPE = CC_EVAL
 INSTALL_METHOD = overwrite
 TCB = yes
 DESKTOP = NONE vagy CDE
 ENABLE_64BIT_KERNEL = yes
 CREATE_JFS2_FS = yes
 ALL_DEVICES_KERNELS = no
 FIREFOX_BUNDLE = no
 HTTP_SERVER_BUNDLE = no
 KERBEROS_5_BUNDLE = no
 SERVER_BUNDLE = no
 ALT_DISK_INSTALL_BUNDLE = no

locale:
 CULTURAL_CONVENTION = en_US vagy C
 MESSAGES = en_US vagy C
```

A NIM vezérlő nem állítható be BAS/EAL4+ rendszerként és nem csatlakoztatható ugyanahhoz a hálózathoz más BAS/EAL4+ rendszerekkel. Amikor kezdeményezi a telepítést a NIM vezérlőről, akkor a **NIM ügyfél maradjon a SMIT telepítése után** menüpontot Nem értékűre kell állítani. Miután telepítette a NIM ügyfelet BAS/EAL4+ rendszerként, a NIM ügyfelet el kell távolítani a NIM vezérlő hálózatából, és nem végezhető további szoftvertelepítés vagy -frissítés a NIM vezérlő használatával.

Például tegyük fel, hogy két hálózati környezettel rendelkezik; az egyik hálózat a NIM vezérlőből és a nem BAS/EAL4+ rendszerekből áll, a másik hálózat pedig csak BAS/EAL4+ rendszerekből áll. Végezze el a NIM telepítést a NIM ügyfélen. A telepítés befejeződése után kapcsolja le az újonnan telepített BAS/EAL4+ rendszert a NIM vezérlő hálózatáról és csatlakoztassa azt a kiértékelt hálózathoz.

A második példa egyetlen hálózatot tartalmaz. A NIM vezérlő nem csatlakozik a hálózathoz, amikor a többi rendszer működik a kiértékelt konfigurációban, és a BAS/EAL4+ rendszerek nem csatlakoznak a hálózathoz a NIM telepítés során.

BAS/EAL4+ szoftvercsomag:

Amikor kiválasztja a **BAS/EAL4+** lehetőséget, akkor telepítésre kerül a /usr/sys/inst.data/sys_bundles/CC_EVAL.BOS.autoi telepítési szoftvercsomag tartalma.

Nem kötelezően kiválaszthatja a Grafikai szoftver csomag és a Dokumentációs szolgáltatások szoftver csomag telepítését is a kiválasztott **BAS/EAL4+** lehetőséggel. Ha kiválasztja a **Grafikai szoftver** lehetőséget is a **BAS/EAL4+** lehetőséggel, akkor telepítésre kerül a /usr/sys/inst.data/sys_bundles/CC_EVAL.Graphics.bnd szoftvercsomag tartalma. Ha kiválasztja a Dokumentációs szolgáltatások szoftver lehetőséget a **BAS/EAL4+** lehetőséggel, akkor telepítésre kerül a /usr/sys/inst.data/sys_bundles/CC_EVAL.DocServices.bnd szoftvercsomag tartalma.

A Licencprogram termékek (LPP-k) telepítése után a rendszer módosítja az alapértelmezett konfigurációt, hogy megfeleljen a BAS/EAL4+ követelményeinek. Az alábbi módosítások kerülnek végrehajtásra az alapértelmezett konfiguráción:

- A /dev/echo eltávolítása az /etc/pse.conf fájlból.
- Adatfolyam-eszközök példányosítása.
- Kizárólag root hozzáférés engedélyezése a cserélhető adathordozóhoz.
- Nem CC bejegyzések eltávolítása az dinetd.conf fájlból.
- Különféle fájlengedélyek módosítása.
- Szimbolikus hivatkozások regisztrálása a sysck.cfg fájlba.
- Eszközök regisztrálása a sysck.cfg fájlba.
- Alapértelmezett felhasználó és port attribútumok beállítása.
- A doc_search alkalmazás beállítása a böngészői használatra.
- Httpdlite eltávolítása az inittab fájlból.
- Writesrv eltávolítása az inittab fájlból.
- Mkatmpvc eltávolítása az inittab fájlból.
- Atmsvcd eltávolítása az inittab fájlból.
- Snmpd letiltása az /etc/rc.tcpip fájlban.
- Hostmibd letiltása az /etc/rc.tcpip fájlban.
- Snmpmibd letiltása az /etc/rc.tcpip fájlban.
- Aixmibd letiltása az /etc/rc.tcpip fájlban.
- Muxatmd letiltása az /etc/rc.tcpip fájlban.
- Az NFS port (2049) egy jogosultságot igényló port.
- Hiányzó események felvétele az /etc/security/audit/events fájlba.
- Annak biztosítása, hogy a loopback csatoló fut.
- Szinonimák létrehozása a /dev/console számára.
- Alapértelmezett X-server kapcsolat engedélyek betartatása.
- A /var/docsearch könyvtár módosítása, hogy az összes fájl az egész világra kiterjedően olvasható legyen.
- Object Data Manager (ODM) szakaszok felvétele a konzolengedélyek beállítása érdekében.
- A BSD stílusú programtípusok engedélyeinek 000-ra állítása.
- A .netrc fájlok letiltása.
- Javítási könyvtár feldolgozás hozzáadása.

Grafikus felhasználói felület:

A BAS/EAL4+ szabványnak megfelelő rendszer tartalmazza az X Windows rendszert mint grafikus felhasználói felületet.

Az X Windows lehetővé teszi grafikus ügyfelek, például órák, számológépek és egyéb grafikus alkalmazások, valamint több terminálszekció megjelenítését az **aixterm** paranccsal. Az X Windows rendszert az **xinit** paranccsal lehet elindítani a kezdeti parancssorból, miután a felhasználó bejelentkezett a hoszt konzolján.

X Windows munkamenet indításához írja be a következőt:

```
xinit
```

A parancs úgy indítja el az X Windows kiszolgálót, hogy a helyi hozzáférési mechanizmusok csak a hívó számára engedélyezettek. A root felhasználóra állított UID azonosítóval rendelkező X Windows ügyfelek hozzáférhetnek az X Windows kiszolgálóhoz a UNIX tartomány socket útján a root felülbírálás használatával a hozzáférési korlátozásokra. A más felhasználóra állított UID azonosítóval rendelkező X Windows ügyfelek nem férhetnek hozzá az X Windows kiszolgálóhoz. Ez a korlátozás megakadályozza, hogy egy hoszt más felhasználói jogosulatlan hozzáférést szerezzenek az X Windows kiszolgálóhoz.

LAS /EAL4+ rendszer telepítése:

Az RBAC automatikusan engedélyezett, amikor ez a lehetőség ki van választva.

Ha be akarja állítani a LAS/EAL4+ lehetőséget egy BOS telepítés során, akkor tegye a következőket:

A telepítési beállítások eléréséhez írjon be egy 3-ast a **Biztonsági modell** módosításához, és egy 4-est a **További beállítások** mező megjelenítéséhez a Telepítés és beállítások ablakban. Ezek a beállítások a telepítés típusa (felülírás, megőrzés vagy átállítás) és a biztonsági beállítások alapján változóak. LAS esetén a telepítési módszer új vagy teljes felülírás. Válassza a **LAS/EAL4+ konfiguráció telepítése** lehetőséget.

További RBAC információkat itt talál: Szerep alapú hozzáférés-felügyelet (RBAC).

LAS/EAL4+ konfiguráció telepítése (csak Trusted AIX rendszerrel áll rendelkezésre):

A **LAS/EAL4+ konfiguráció telepítése** lehetőség telepíti a Trusted AIX rendszert LAS/EAL4+ beállított üzemmódban. A LAS/EAL4+ beállított üzemmód további korlátozó biztonsági szolgáltatásokat nyújt a Trusted AIX telepítéshez képest.

Ha csendes telepítést végez egyéni bosinst.data fájl használatával, akkor az **INSTALL_TYPE** mezőnek üresnek kell lennie, a **TRUSTED_AIX** mezőt yes értékűre kell állítani, a következő mezőket pedig az alábbiak szerint kell beállítani:

```
control_flow:
 CONSOLE = ???
 PROMPT = yes
 INSTALL_TYPE =
 TRUSTED_AIX = yes
 INSTALL_METHOD = overwrite
 TCB = yes
 DESKTOP = NONE
 ENABLE_64BIT_KERNEL = yes
 CREATE_JFS2_FS = yes
 ALL_DEVICES_KERNELS = no
 FIREFOX_BUNDLE = no
 HTTP_SERVER_BUNDLE = no
 KERBEROS_5_BUNDLE = no
 SERVER_BUNDLE = no
 ALT_DISK_INSTALL_BUNDLE = no
```

```
locale:
 CULTURAL_CONVENTION = en_US vagy C
 MESSAGES = en_US vagy C
```

A Megbízható AIX rendszerrel kapcsolatos további információkat a Megbízható AIX témakör tartalmaz.

Hálózati telepítéskezelés környezet LAS/EAL4+ esetén:

A LAS/EAL4+ technológia ügyfelek telepítése végrehajtható a Hálózati telepítéskezelés (NIM) környezet használatával.

A NIM vezérlő úgy van beállítva, hogy biztosítsa a szükséges erőforrásokat az AIX 7.1 megfelelő LAS/EAL4+ szintjének telepítéséhez. Ezután a NIM ügyfelek a NIM vezérlőn található erőforrások használatával telepíthetők. Végrehajthatja az ügyfél csendes NIM telepítését az alábbi mezők beállításával a bosinst_data erőforrásban:

```
control_flow:
 CONSOLE = ???
 PROMPT = no
 INSTALL_TYPE =
 TRUSTED_AIX = yes
 INSTALL_METHOD = overwrite
 TCB = yes
 DESKTOP = NONE
 ENABLE_64BIT_KERNEL = yes
 CREATE_JFS2_FS = yes
 ALL_DEVICES_KERNELS = no
 FIREFOX_BUNDLE = no
 HTTP_SERVER_BUNDLE = no
 KERBEROS_5_BUNDLE = no
 SERVER_BUNDLE = no
 ALT_DISK_INSTALL_BUNDLE = no

locale:
 CULTURAL_CONVENTION = en_US vagy C
 MESSAGES = en_US vagy C
```

A NIM vezérlő nem állítható be LAS/EAL4+ rendszerként és nem csatlakoztatható ugyanahhoz a hálózathoz más LAS/EAL4+ rendszerekkel. Amikor kezdeményezi a telepítést a NIM vezérlőről, akkor a **NIM ügyfél maradjon a SMIT telepítése után** menüpontot Nem értékűre kell állítani. Miután telepítette a NIM ügyfelet LAS/EAL4+ rendszerként, a NIM ügyfelet el kell távolítani a NIM vezérlő hálózatából, és nem végezhető további szoftvertelepítés vagy -frissítés a NIM vezérlő használatával.

Például tegyük fel, hogy két hálózati környezettel rendelkezik; az egyik hálózat a NIM vezérlőből és a nem LAS/EAL4+ rendszerekből áll, a másik hálózat pedig csak LAS/EAL4+ rendszerekből áll. Végezze el a NIM telepítést a NIM ügyfélen. A telepítés befejeződése után kapcsolja le az újonnan telepített LAS/EAL4+ rendszert a NIM vezérlő hálózatáról és csatlakoztassa azt a kiértékelt hálózathoz.

A második példa egyetlen hálózatot tartalmaz. A NIM vezérlő nem csatlakozik a hálózathoz, amikor a többi rendszer működik a kiértékelt konfigurációban, és a LAS/EAL4+ rendszerek nem csatlakoznak a hálózathoz a NIM telepítés során.

BAS/EAL4+ és LAS/EAL4+ rendszerek fizikai környezete:

A BAS/EAL4+ és LAS/EAL4+ rendszerek specifikus követelményeket támasztanak a futtató környezettel szemben.

A követelmények az alábbiak:

- Korlátozni kell a fizikai hozzáférést a rendszerekhez, hogy csak a jogosult adminisztrátorok használhassák a rendszerkonzolokat.
- A szervizprocesszor nem csatlakozik modemhez.

- A terminálok fizikai hozzáférése a jogosult felhasználókra korlátozott.
- A fizikai hálózat védett a lehallgatási és IP cím hamisítási programok (úgynevezett trójai faló) programok ellen. A nem védett vonalakon folytatott kommunikációhoz további biztonsági intézkedések, például titkosítás szükséges.
- Nem megengedett a kommunikáció más olyan rendszerekkel, amelyek nem AIX 7.1 BAS/EAL4+ vagy LAS/EAL4+ rendszerek, vagy nem ugyanazon felügyeleti vezérlés alatt vannak.
- Csak IPv4 protokoll használható a más BAS/EAL4+ és LAS/EAL4+ rendszerekkel folytatott kommunikációhoz. Az IPv6 protokollt tartalmazza a kiértékelt konfiguráció, de az IPv6 protokollnak csak azokat a működési képességeit, amelyeket az működő is támogat.
- A felhasználóknak nem szabad engedélyezni a rendszeridő módosítását.
- LPAR környezetben található rendszerek nem osztozhatnak PHB-ken.

BAS/EAL4+ és LAS/EAL4+ rendszer szervezeti környezete:

BAS/EAL4+ és LAS/EAL4+ rendszerek esetén teljesülniük kell bizonyos eljárásbeli és szervezeti követelményeknek.

Az alábbi követelményeknek kell teljesülniük:
- Az adminisztrátoroknak megbízható, szakképzett személyeknek kell lenniük.
- Csak a rendszeren található információk kezelésére jogosult felhasználók kaphatnak felhasználói azonosítót a rendszerhez.
- A felhasználók jó minőségű jelszókat kell használniuk (amennyire lehetséges, véletlenszerű értékeket, amelyek nem kapcsolódnak a felhasználóhoz vagy a szervezethez). A jelszószabályok beállításával kapcsolatos információkat itt talál: “Jelszavak” oldalszám: 62.
- A felhasználók nem adhatják ki a jelszavukat másoknak.
- Az adminisztrátoroknak elegendő ismerettel kell rendelkezniük az olyan rendszerek kezelésével kapcsolatban, ahol kritikus a biztonság.
- Az adminisztrátoroknak a rendszer dokumentációjában biztosított iránymutatás szerint kell végezniük a munkájukat.
- Az adminisztrátoroknak saját nevükön és jelszavukkal kell bejelentkezni, majd az **su** parancs segítségével válthatnak superuser módra az adminisztrációhoz.
- Az adminisztrátorok által a rendszerfelhasználók számára előállított jelszókat biztonságosan kell továbbítani a felhasználók számára.
- A rendszerért felelős személyeknek ki kell alakítaniuk és meg kell valósítaniuk a szükséges eljárásokat a rendszerek biztonságos működéséhez.
- Az adminisztrátoroknak biztosítaniuk kell, hogy a biztonsági szempontból kritikus rendszererőforrásokat védjék az engedély bitek és ACL listák megfelelő beállításai.
- A szervezetnek jóvá kell hagynia a fizikai hálózatot, hogy az szállíthassa a rendszereken található legérzékenyebb adatokat.
- A karbantartási eljárásoknak tartalmazniuk kell a rendszerek rendszeres diagnosztizálását.
- Az adminisztrátoroknak rendelkezniük kell megfelelő eljárásokkal a biztonságos működés és a rendszer meghibásodás utáni helyreállítás biztosításához.
- A *LIBPATH* környezeti változót nem szabad módosítani, mivel ez azt eredményezheti, hogy egy megbízható folyamat egy megbízhatatlan függvénytárat tölt be.
- Működő rendszeren nem használható lehallgatási és nyomkövetési szoftver (tcpdump, trace).
- Névtelen protokollok (például HTTP) csak nyilvános információkhoz használhatók (például online dokumentációhoz).
- Csak TCP alapú NFS használható.
- A felhasználóknak nem szabad hozzáférést adni a cserélhető adathordozókhoz. Az eszközfájlokat a megfelelő engedély bitekkel vagy ACL listákkal kell védeni.
- Az adminisztrátorok nem használhatnak dinamikus particionálást az erőforrások kiosztásához, illetve kiosztásának megszüntetéséhez. Csak akkor lehet partíciókonfigurálást végezni, amikor egyetlen partíció sem fut.

BAS/EAL4+ és LAS/EAL4+ rendszer működési környezete:

BAS/EAL4+ és LAS/EAL4+ rendszerek esetén teljesülniük kell bizonyos működési követelményeknek és eljárásoknak.

Az alábbi követelményeknek és eljárásoknak kell teljesülniük:

- Hardware Management Console (HMC) használata esetén a HMC egy fizikailag vezérelt környezetben található.
- Csak arra jogosult személy rendelkezik működési környezet és HMC hozzáféréssel.
- HMC használata esetén a HMC csak az alábbi feladatokhoz használható:
  - A partíciók kezdeti beállítása. A partíció nem lehet aktív a beállítási folyamat során.
  - "Lefagyott" partíciók újraindítása
- A HMC nem használható a beállított rendszer működése során.
- A rendszer "hazahívás" szolgáltatását le kell tiltani.
- A távoli modemes hozzáférést a rendszerhez le kell tiltani.
- Ha az AIX egy LPAR támogatással rendelkező környezetben fut, akkor az adminisztrátor az LPAR dokumentációban ellenőrizze a logikai partíciók EAL4+ működésére vonatkozó követelményeket.
- A szervizjogosultság szolgáltatást le kell tiltani a logikai partíciókon.

BAS/EAL4+ rendszerkonfiguráció:

Beállíthatja a Base AIX Security (BAS) és Evaluation Assurance Level 4+ (EAL4+) rendszert.

A **system**, **sys**, **adm**, **uucp**, **mail**, **security**, **cron**, **printq**, **audit** és **shutdown** csoportokat a rendszer adminisztrátori csoportoknak tekinti. Csak megbízható felhasználókat szabad hozzáadni ehhez a csoporthoz.

Adminisztráció:

Az adminisztrátoroknak a személyes felhasználói fiókjukkal kell bejelentkezniük, majd a **su** paranccsal a root felhasználóra váltaniuk a rendszer adminisztrálása céljából.

A root fiók jelszavának hatékony védelme érdekében csak jogosult adminisztrátoroknak engedélyezze a **su** parancs használatát a root fiókra. Ennek biztosításához tegye a következőket:

1. Vegyen fel egy bejegyzést az /etc/security/user fájl **root** szakaszába az alábbiak szerint:

```
root:
  admin = true
  .
  .
  .
  sugroups = SUADMIN
```

2. Határozzon meg egy olyan csoportot az /etc/group fájlban, amely csak a jogosult adminisztrátorok felhasználói azonosítóját tartalmazza, az alábbiak szerint:

```
system:!:0:root,paul
staff:!:1:invscout,julie
bin:!:2:root,bin
.
.
.
SUADMIN:!:13:paul
```

Az adminisztrátoroknak is ragaszkodniuk kell az alábbiakhoz:

- Ki kell alakítani és meg kell valósítani eljárásokat annak biztosítása érdekében, hogy az osztott rendszert alkotó hardver, szoftver és firmware összetevők védett módon legyenek terjesztve, telepítve és konfigurálva.
- Gondoskodjék róla, hogy a rendszer úgy legyen konfigurálva, hogy csak adminisztrátor vezethessen be új, megbízható szoftvert a rendszerbe.

- Valósítson meg eljárásokat annak érdekében, hogy a felhasználók kiürítsék a képernyőt, mielőtt kijelentkeznek a soros bejelentkezési eszközökről (például IBM® 3151 terminálokról).

Felhasználó- és portkonfiguráció:

Az AIX felhasználó- és portkonfigurációját úgy kell beállítani, hogy megfeleljen a kiértékelési követelményeknek. A tényleges követelmény, hogy a TSF olyan mechanizmust biztosítson a jelszó kitalálása ellen, amely megfelel a mérési minőségnek. Annak valószínűsége, hogy egy támadó helyesen kitalálja a jelszót annak élete során, legyen kisebb, mint 2^-20.

Az alábbi példa /etc/security/user fájlja az /usr/share/dict/words szótárlistát használja. Az /usr/share/dict/words fájl a bos.data fájlkészlet része. Az /etc/security/user fájl beállításának megkezdése előtt telepíteni kell a bos.data fájlkészletet. Az /etc/security/user fájl javasolt értékei tehát a következők:

```
default:
  admin = false
  login = true
  su = true
  daemon = true
  rlogin = true
  sugroups = ALL
  admgroups =
  ttys = ALL
  auth1 = SYSTEM
  auth2 = NONE
  tpath = nosak
  umask = 077
  expires = 0
  SYSTEM = "compat"
  logintimes =
  pwdwarntime = 5
  account_locked = false
  loginretries = 3
  histexpire = 52
  histsize     = 20
  minage = 0
  maxage = 8
  maxexpired = 1
  minalpha = 2
  minother = 2
  minlen = 8
  mindiff = 4
  maxrepeats = 2
  dictionlist = /usr/share/dict/words
  pwdchecks =
  dce_export = false

root:
  rlogin = false
  login = false
```

Az /etc/security/user fájl alapértelmezett beállításait nem szabad felülírni egyes felhasználók egyedi beállításaival.

Megjegyzés: A root szakaszba beírt login = false megakadályozza a közvetlen rootként bejelentkezést. Csak a root fiókra nézve **su** jogosultsággal rendelkező felhasználói fiókok jelentkezhetnek be root fiókként. Ha a rendszer ellen DoS típusú támadás indul, amely helytelen jelszavakat küld a felhasználói fiókokra, akkor előfordulhat, hogy az összes felhasználói fiók zárolódik. Ez a támadás megakadályozhatja a felhasználók (közöttük az adminisztrátorok) bejelentkezését a rendszerbe. Ha a felhasználói fiók zárolva van, akkor a felhasználó addig nem tud belépni, amíg a rendszeradminisztrátor a /etc/security/lastlog fájlban vissza nem állítja a felhasználó unsuccessful_login_count attribútumát a loginretries felhasználói attribútumnál kisebb értékre. Ha az összes adminisztrátori fiók zárolódott, akkor lehet, hogy újra kell indítani a rendszert karbantartási módban, és futtatni kell a **chsec** parancsot. Ha további tájékoztatásra van szüksége a **chsec** parancs használatával kapcsolatban, olvassa el: “Felhasználói fiók felügyelet” oldalszám: 51.

A /etc/security/login.cfg fájl javasolt értékei a következők:

```
default:
  sak_enabled = false
  logintimes =
  logindisable = 4
  logininterval = 60
  loginreenable = 30
  logindelay = 5
```

setuid/setgid programok listája:

Létrehozásra kerül a megbízható alkalmazások listája a BAS támogatással rendelkező AIX rendszerekhez.

A **suid/sgid** bit kikapcsolásra kerül minden olyan nem megbízható program esetén, amelynek a root felhasználó vagy egy megbízható csoport a tulajdonosa. BAS telepítés után a rendszeren csak olyan programok vannak, amelyek vagy **suid** és tulajdonosuk root, vagy **sgid** és tulajdonosuk az alábbi megbízható csoportok egyike: **system**, **sys**, **adm**, **uucp**, **mail**, **security**, **cron**, **printq**, **audit** és **shutdown**. Csak megbízható felhasználóhat vegyen fel ezekbe a csoportokba.

A megbízható alkalmazások listája az összes olyan alkalmazás figyelembe vételével kerül összeállításra, amelyek az alábbi kategóriák közül legalább egybe beletartoznak:

- A vonatkozó alkalmazáshoz tartozó SUID root bit engedélyezett
- Az SGID bit a megbízható csoportok egyikére engedélyezett
- Azok az alkalmazások, amelyek hozzáférnek a megbízható adatbázisok valamelyikéhez az adminisztrátori segítségnyújtó dokumentum szerint

Megjegyzés: A rendszeradminisztrátornak el kell távolítania az **ipcs** parancshoz tartozó **setuid** bitet. A rendszeradminisztrátornak futtatnia kell a **chmod u-s /usr/bin/ipcs** és a **chmod u-s /usr/bin/ipcs64** parancsot.

Felülvizsgálati fájlrendszer módosítása:

Az RBAC automatikusan engedélyezett, amikor ez a lehetőség ki van választva.

Az /audit fájlrendszer jfs fájlrendszer. Módosítani kell jfs2 fájlrendszerre. A BAS rendszerek további parancsokat is igényelnek. A fájlrendszer módosításához tegye a következőket:

1. A BAS rendszerek fájlrendszerének módosításához adja ki az alábbi parancsot:

   ```
   audit shutdown
   lsvg -l rootvg
   ```

   LAS rendszerek esetén folytassa a 3. lépéssel.
2. Ha a TYPE mező kérdőjelet (?) tartalmaz, akkor adja ki az alábbi parancsot:

   ```
   synclvodm -v rootvg
   ```
3. Távolítsa el a jfs fájlrendszert és hozzon létre jfs2 fájlrendszert az alábbi parancs kiadásával:

   ```
   umount/audit
   rmfs /audit
   crfs -v jfs2 -m /audit -g rootvg -A yes -p rw -a size=100M
   ```

Megbízható aláírás-adatbázis (TSD) frissítése:

A szakasz ismerteti a TSD frissítésének módját.

A BAS/LAS konfiguráció módosítja a rendszer üzemmód bitjeit, amely TSD integritási hibához vezet.

A rendszer újraindítása során válassza az **Összes mellőzése** lehetőséget.

A TSD frissítéséhez írja be az alábbi parancsot:

```
trustchk -u ALL mode
```

LAS rendszer használata:

Ez a rész irányelveket biztosít a LAS rendszer használatához.

Állítsa az automatikus újraindítás paramétert **false** értékűre, miután telepítette a rendszert isso-ként. Ehhez adja ki a következő parancsot:

```
chdev -l sys0 -a autorestart=false
```

Ha a TSD továbbra is intlabel hibákat állít elő, akkor a hibák törléséhez használja az isso-t **PV_ROOT** jogosultsággal, és adja ki az alábbi parancsokat:

```
cp /etc/security/tsd/tsd.dat /etc/security/tsd/tsd.dat.org
trustchk -q /usr/sbin/format /usr/sbin/fdformat /usr/sbin/mount /usr/sbin/unmount \
/usr/sbin/umount /usr/sbin/tsm /usr/sbin/getty /usr/sbin/login /usr/sbin/mkvg \
/usr/sbin/extendvg /usr/bin/w /usr/bin/uptime >/tmp/list.dat
grep -p SLTL /tmp/list.dat |sed 's/SLTL/SHTL/' >/tmp/new.dat
trustchk -w -a -f /tmp/new.dat
trustchk -y ALL
```

Ha a felülvizsgálathoz kapcsolódó hibaüzenetek jelennek meg a konzolon, akkor isso jogosultsággal indítsa újra a felülvizsgálati rendszert az alábbi parancsok kiadásával:

```
# audit shutdown
# audit start
```

Három sikertelen bejelentkezési kísérlet után a rendszer letiltja az isso/so bejelentkezést. Az adminisztrátor azonban továbbra is hozzáférhet ezekhez a fiókokhoz a helyi konzolon.

A cron/at által futtatott parancsok kimenete nem kerül továbbításra a felhasználó levelezési sorába.

A világ számára írható könyvtárak, amelyek címketartománnyal rendelkeznek (például /tmp), nem particionáltak. A címkék közötti információfolyam lehetőségének megakadályozása érdekében az adminisztrátornak particionálnia kell ezeket a könyvtárakat közvetlenül a kezdeti beállítás után.

Hálózati csatoló:

A szakasz ismerteti a hálózati csatoló használatának módját.

Trusted AIX rendszeren az alapértelmezett hálózati csatoló a minSL=impl_lo és maxSL=ts_all címketartománnyal rendelkezik. LAS/EAL4+ rendszerek esetén nincs címketartomány. Az alapértelmezett szabály automatikusan az impl_lo lesz, amikor kiválasztja a LAS/EAL4+ telepítési beállítást. Ha módosítani akarja az alapértelmezett szabályt isso-ra, akkor adja ki a **netrule** parancsot.

Például:

```
/usr/sbin/netrule i+u default +impl_lo +impl_lo +impl_lo
```

WPAR frissítése:

Ez a szakasz ismerteti, hogy hogyan teheti az EAL4+ szabványnak megfelelővé az AIX munkapartíciókat (WPAR).

Hozza létre a WPAR partíciót BAS rendszeren, és futtassa az alábbi parancsot a WPAR partíción, hogy az EAL4+ szabványnak megfelelővé tegye:

```
/usr/lib/security/CC_EVALify.sh
```

Amikor először futtatja a clogin parancsot egy LAS rendszeren, akkor lefutnak az első rendszerindításra vonatkozó parancsfájlok (például CC_EVALify.sh).

Az első rendszerindításra vonatkozó parancsfájlok miatt a clogin futtatása a szokásosnál tovább tart, mert a clogin meghívja a TSM-et a bejelentkezéshez. Mivel azonban a WPAR továbbra is konfiguráció üzemmódban van, a bejelentkezés meghiúsul. Várjon körülbelül 10 percet, hogy a WPAR befejezze a konfigurálást, csak utána tegyen újabb clogin kísérletet. Újonnan létrehozott WPAR rendszerek esetén az alapértelmezett felhasználó beállításait úgy kell megadni, hogy megfeleljen a kiértékelési követelményeknek, amelyek az alábbiakat tartalmazzák:

- root BAS üzemmódban
- isso/sa/so LAS üzemmódban

A root és az isso felhasználó nem rendelkezik jelszóval vagy gyenge jelszót igényel. A jelszókat frissíteni kell, mielőtt engedélyezné, hogy megbízhatatlan felhasználók hozzáférhessenek a globális környezethez vagy a vonatkozó WPAR partícióhoz.

A kiértékelési jelszó követelmény az, hogy egy jelszó véletlen kitalálási valószínűsége legfeljebb 1:1000000 legyen, és ez ismételt kísérletekkel se növekedhessen egy percen belül 1:100000 fölé. A követelmény kielégítése érdekében a felhasználói paraméterek az /etc/security/user fájlban az alábbiakra változnak:

```
default:
 maxage = 8
 maxexpired = 1
 minother = 2
 minlen = 8
 maxrepeats = 2
 loginretries = 3
 histexpire = 52
 histsize     = 20
```

EFS frissítése:

A szakasz ismerteti, hogy hogyan állíthatja be a kriptográfiai fájlrendszerként kiértékelt EFS biztonsági attribútumait.

A kiértékelés nem tartalmazza a root védelmi üzemmód szempontjait a teljes root hozzáféréssel szemben. Az EFS engedélyezésekor állítsa be az **efsmgr** és **egskeymgr** parancsok biztonsági attribútumait az alábbi parancs kiadásával:

```
setsecattr -c accessauths=ALLOW_ALL
innateprivs=PV_DEV_QUERY,PV_DEV_CONFIG,PV_AU_ADD,PV_DAC_R,PV_DAC_W,PV_DAC_X /usr/sbin/efsmgr

setsecattr -c accessauths=ALLOW_ALL
innateprivs=PV_DEV_QUERY,PV_DEV_CONFIG,PV_AU_ADD,PV_DAC_R,PV_DAC_W,PV_DAC_X /usr/sbin/efskeymgr

setkst -t cmd
```

Merevlemez törlése:

Az AIX lehetővé teszi a merevlemezek törlését az AIX diagnosztikai csomagban található **Adathordozó formázása** szervizsegédlet használatával. A diagnosztikai csomag teljes dokumentációja a *Többsínes rendszerek diagnosztikai információi* című könyvben és a hardver felhasználói kézikönyvében található.

Merevlemez törléséhez futtassa a következő parancsot:

```
diag -T "format"
```

A parancs elindítja az **Adathordozó formázása** szervizsegédletet egy menüvezérelt felületen. Amikor a rendszer rákérdez, válassza ki a terminált.

Megjelenik egy erőforrás-kiválasztási lista. Válassza ki a listából a törölni kívánt merevlemez-eszközöket, és véglegesítse a módosításokat a képernyőn látható útmutatás szerint.

A választások véglegesítése után válassza a **Lemez törlése** menüpontot. Ezután a rendszer megkéri, hogy erősítse meg a választását. Kattintson az **Igen** gombra.

Ezután választania kell az **Adatok beolvasása a meghajtóról** vagy a **Minták írása a meghajtóra** lehetőségek közül. Válassza a **Minták írása a meghajtóra** lehetőséget.

Ezután ha akarja, módosíthatja a lemeztörlési beállításokat. A kívánt beállítások megadása után kattintson a **Módosítások véglegesítése** gombra. A lemez törlésre kerül.

Megjegyzés: A folyamat hosszú időbe telhet.

Erőforrásokra vonatkozó korlátozások:

Amikor erőforrásokra vonatkozó korlátozásokat állít be az /etc/security/limits fájlban, akkor gondoskodjék róla, hogy a korlátozások illeszkedjenek a rendszeren futó folyamatok igényeihez.

Főleg soha ne állítsa be a stack méretet unlimited értékre. A korlátlan verem felülírhatja a futó folyamat más szegmenseit. A stack_hard méretének szintén korlátozottnak kell lennie.

Felülvizsgálati alrendszer:

Számos olyan eljárás van, amely segíti a felülvizsgálati alrendszer védelmét.

- A felülvizsgálati alrendszert úgy kell beállítani, hogy rögzítse a felhasználók összes releváns biztonsági tevékenységét. Annak biztosításához, hogy a felülvizsgálathoz szükséges fájlterület elérhető és nem használja más fájlrendszer-terület fogyasztó, állítson be kijelölt fájlrendszert a felülvizsgálati adatok számára.
- Védje a felülvizsgálati rekordokat (például a megfigyelési nyomkövetéseket, a bin fájlokat és az /audit alkönyvtárban tárolt összes adatot) a nem root felhasználóktól.
- BAS/EAL4+ rendszer esetén **bin** üzemmódú felülvizsgálatot kell beállítani a felülvizsgálati alrendszer használatakor. A felülvizsgálati alrendszer beállításával kapcsolatos információkat itt talál: “Megfigyelés beállítása” oldalszám: 144.
- A rendszeren elérhető szabad lemezterületnek legalább a 20 százalékát a megfigyelési nyomkövetés számára kell fenntartani.
- Ha a felülvizsgálat engedélyezett, akkor az /etc/security/audit/config fájl start szakaszában a *binmode* paramétert panic értékűre kell állítani. A bin szakasz *freespace* paraméterét legalább a megfigyelési naplók tárolására szánt lemezterület 25 százalékára kell állítani. A *bytethreshold* és a *binsize* paramétert 65 536 byte-ra kell állítani.
- Másolja át a rendszerről a felülvizsgálati rekordokat megőrzött tárolóba archiválás céljából.

Nem megosztott fájlok az osztott rendszerben:

Az /etc/security könyvtár alábbi fájljait nem kell megosztani az osztott rendszerben, hanem a hosztra jellemzőnek kell maradniuk.

/etc/security/failedlogin
A meghiúsult bejelentkezések naplófájlja hosztonként

/etc/security/lastlog
Felhasználónkénti információk a legutóbbi sikeres és sikertelen bejelentkezésről az adott hoszton

/etc/security/login.cfg
Hosztspecifikus bejelentkezési jellemzők: megbízható elérési út, bejelentkezési parancsértelmezők és egyéb, bejelentkezéssel kapcsolatos információk

/etc/security/portlog
Portonkénti információk a zárolt portokról a hoszton

A megosztott fájlok automatikusan előállított biztonsági mentési fájljai szintén nem megosztottak. A biztonsági mentési fájlok neve ugyanaz, mint az eredeti fájloké, csak eléjük van téve egy kis o betű.

DACinet szolgáltatás használata a felhasználó alapú és a port alapú hálózati hozzáférés-felügyelethez:

A DACinet szolgáltatással korlátozható a felhasználók hozzáférése a TCP portokhoz.

A DACinet szolgáltatással kapcsolatos további információkat itt talál: “Felhasználó alapú TCP port hozzáférés felügyelet az internet portok kizárólagos hozzáférés felügyeletével” oldalszám: 207. Például amikor a DACinet szolgáltatással korlátozza a TCP/25 bejövő port elérését a root felhasználóra, akkor csak a root felhasználók érhetik el ezt a portot BAS/EAL4+ szabványnak megfelelő hosztokról. Ez a helyzet korlátozza annak lehetőségét, hogy a normál felhasználók IP cím hamisítást kövessenek el az e-mail szolgáltatáson úgy, hogy telnet használatával csatlakoznak az áldozat TCP/25 portjához.

Ahhoz, hogy aktiválja az ACL listákat a TCP kapcsolatokhoz a rendszerbetöltéskor, futtassa az /etc/rc.dacinet parancsfájlt az /etc/inittab helyről. Ez beolvassa az /etc/security/acl fájlban található meghatározásokat és betölti az ACL listákat a kernelbe. Azokat a portokat, amelyeket nem kell ACL listákkal védeni, az /etc/security/services fájlban sorolja fel. Ez ugyanolyan formátumú, mint az /etc/services fájl.

Feltéve, hogy az összes csatlakozó rendszer a 10.1.1.0/24 alhálózaton található, az X (TCP/6000) portról a root felhasználó kivételével mindenkit kitiltó ACL bejegyzés a következőképpen nézne ki az /etc/security/acl fájlban:

```
6000    10.1.1.0/24 u:root
```

További szoftver telepítése BAS/EAL4+ szabványnak megfelelő rendszerre:

Az adminisztrátor további szoftvert telepíthet a BAS/EAL4+ szabványnak megfelelő rendszerre. Ha a szoftvert nem a root felhasználó vagy root felhasználói jogosultságokkal rendelkező felhasználó futtatja, az nem érvényteleníti a BAS/EAL4+ megfelelést. Jellemző példa az olyan irodai alkalmazások esete, amelyeket csak normál felhasználók futtatnak és nem rendelkeznek SUID összetevőkkel.

Továbbá a root felhasználói jogosultságokkal futó telepített szoftver érvényteleníti a BAS/EAL4+ megfelelést. Ez azt jelenti, hogy például a régebbi JFS illesztőprogramokat nem szabad telepíteni, mivel ezek kernel üzemmódban futnak. Az olyan alkalmazások, amelyeknek egy vagy több jogosultságot adott az /etc/security/privcmds útján, nem elfogadhatóak. A root-ként futó egyéb démonok (például az SNMP démon) is érvénytelenítik a BAS/EAL4+ megfelelést. A BAS/EAL4+ támogatással rendelkező rendszer nem frissíthető (normál esetben).

A BAS/EAL4+ szabványnak megfelelő rendszer ritkán használatos a kiértékelt konfigurációban, különösen üzleti környezetben. Jellemzően további szolgáltatások szükségesek, hogy az éles rendszer alapja egy kiértékelt rendszer legyen, de nem felel meg a kiértékelt rendszer pontos specifikációjának.

NSF v4 hozzáférés-felügyeleti listák és tartalom irányelv:

Az NFS v4 hozzáférés-felügyeleti lista (ACL) **Típus**, **Maszk** és **Kapcsolók** mezőt tartalmaz.

A mezők leírása:

- A **Típus** mező az alábbi értékek egyikét tartalmazza:
  - ALLOW – Megadja a **Ki** mezőben megadott alanyoknak a **Maszk** mezőben megadott jogosultságo(ka)t.
  - DENY – Megtagadja a **Ki** mezőben megadott alanyoktól a **Maszk** mezőben megadott jogosultságo(ka)t.
- A Maszk mező az alábbi, finoman szabályozott jogosultság értékek közül tartalmaz legalább egyet:
  - READ_DATA / LIST_DIRECTORY – Az adatok beolvasása egy nem-könyvtár objektumból vagy egy könyvtárban található objektumok listázása.
  - WRITE_DATA / ADD_FILE – Adatok írása egy nem-könyvtár objektumba vagy nem-könyvtár objektum felvétele egy könyvtárba.
  - APPEND_DATA / ADD_SUBDIRECTORY – Adatok hozzáfűzése egy nem-könyvtár objektumhoz vagy alkönyvtár hozzáadása egy könyvtárhoz.
  - READ_NAMED_ATTRS – Egy objektum nevesített attribútumainak beolvasása.
  - WRITE_NAMED_ATTRS – Egy objektum nevesített attribútumainak írása.

- EXECUTE – Fájl futtatása vagy egy könyvtár bejárása, illetve keresés egy könyvtárban.
- DELETE_CHILD – Fájl vagy könyvtár törlése egy könyvtáron belül.
- READ_ATTRIBUTES – Egy fájl alapvető (nem ACL) attribútumainak beolvasása.
- WRITE_ATTRIBUTES – Egy fájlhoz vagy könyvtárhoz tartozó időpontok módosítása.
- DELETE – Fájl vagy könyvtár törlése.
- READ_ACL – Az ACL beolvasása.
- WRITE_ACL – Az ACL írása.
- WRITE_OWNER – A tulajdonos és csoport módosítása.
- SYNCHRONIZE – Hozzáférés szinkronizálása (a többi NFS v4 ügyféllel való kompatibilitás érdekében létezik, de nincs megvalósított funkciója)

- **Kapcsolók** mező - A mező meghatározza a könyvtár ACL listák átörökítési képességeit, valamint jelzi, hogy a **Ki** mező tartalmaz-e csoportot vagy sem. A mező nulla vagy több kapcsolót tartalmaz az alábbiak közül:
  - **FILE_INHERIT** – Megadja, hogy a könyvtárban az újonnan létrehozott nem-könyvtár objektumok öröklik-e ezt a bejegyzést.
  - **DIRECTORY_INHERIT** – Megadja, hogy a könyvtárban az újonnan létrehozott alkönyvtárak öröklik-e ezt a bejegyzést.
  - **NO_PROPAGATE_INHERIT** – Megadja, hogy a könyvtárban az újonnan létrehozott alkönyvtárak öröklik ezt a bejegyzést, de ezek az alkönyvtárak nem adják át ezt a bejegyzést az újonnan létrehozott alkönyvtáraiknak.
  - **INHERIT_ONLY** – Megadja, hogy ez a bejegyzés nem vonatkozik a könyvtárra, csak az újonnan létrehozott objektumokra, amelyek öröklik a bejegyzést.
  - **IDENTIFIER_GROUP** – Megadja, hogy a **Ki** mezőben egy csoport szerepel; egyébként a **Ki** mezőben egy felhasználó vagy egy speciális **Ki** érték van.
- **Ki** mező - A mező az alábbi értékek egyikét tartalmazza:
  - Felhasználó – Megadja azt a felhasználót, akire a bejegyzés vonatkozik.
  - Csoport – Megadja azt a csoportot, amelyre a bejegyzés vonatkozik.
  - Speciális – Ez az attribútum az alábbi értékek egyike lehet:
    - OWNER@ – Megadja, hogy a bejegyzés az objektum tulajdonosára vonatkozik.
    - GROUP@ – Megadja, hogy a bejegyzés az objektumot birtokló csoportra vonatkozik.
    - EVERYONE@ – Megadja, hogy a bejegyzés a rendszer összes felhasználójára vonatkozik, beleértve a tulajdonost és a csoportot is.

Ha az ACL üres, akkor csak a 0 tényleges UID azonosítóval rendelkező alany férhet hozzá az objektumhoz. Az objektum tulajdonosa implicit módon rendelkezik az alábbi maszk értékekkel, függetlenül attól, hogy mit tartalmaz vagy mint nem az ACL:

- READ_ACL
- WRITE_ACL
- READ_ATTRIBUTES
- WRITE_ATTRIBUTES

Az APPEND_DATA mint WRITE_DATA van megvalósítva. Valójában nincs működésbeli különbség a WRITE_DATA és az APPEND_DATA érték között. Az értékeket egymással összhangban kell beállítani vagy beállításukat megszüntetni.

Az objektumok tulajdonjogát a WRITE_OWNER érték használatán keresztül lehet módosítani. A tulajdonos vagy csoport módosításakor a **setuid** bit kikapcsolásra kerül. Az átörökítés kapcsolóknak csak könyvtárak ACL listájában van értelmük és csak azokra az objektumokra vonatkozik, amelyek az átörökítés kapcsoló beállítása után lettek létrehozva a könyvtárban (vagyis a meglévő objektumokra nincs hatással a szülő könyvtár ACL listájában szereplő átörökítés beállítás módosítása). Az NFS v4 ACL bejegyzései sorrendfüggőek. Annak eldöntéséhez, hogy a kért hozzáférés engedélyezett-e, a rendszer sorban dolgozza fel a bejegyzéseket. Csak az alábbi értékkel rendelkező bejegyzéseket veszi figyelembe:

- **Ki** mező, amely megfelel a tényleges UID azonosítónak
- A bejegyzésben vagy a tényleges GID azonosítóval megadott felhasználó
- A tárgy bejegyzésében megadott csoport

A rendszer egyeséve l feldolgozza a bejegyzéseket mindaddig, amíg a kérő hozzáférésének minden bitje ALLOWED. Miután egy bejegyzés engedélyezett egy hozzáférési típust, a rendszer a későbbi bejegyzések feldolgozásakor azt már nem veszi figyelembe. Ha DENY bejegyzés található ott, ahol a kérő hozzáférése az adott maszk értékhez szükséges és nem meghatározott, akkor a kérés megtagadásra kerül. Ha a kiértékelés eléri az ACL végét, akkor a kérés megtagadásra kerül.

Az ACL támogatott maximális mérete 64 KB. Az ACL egyes bejegyzései változó hosszúságúak, csak a 64 KB-os korlát érvényes.

WRITE OWNER érték:

Az NFS v4 irányelv biztosítja annak vezérlését, hogy ki olvashatja és írhatja az objektumok attribútumait.

A 0 tényleges UID azonosítóval rendelkező alany mindig felülbírálhatja az NFS v4 irányelvet. Az objektum tulajdonosa az ACL maszk READ_ATTRIBUTES, WRITE_ATTRIBUTES, READ_NAMED_ATTRS és WRITE_NAME_ATTRS attribútumaival engedélyezheti mások számára az objektum attribútumainak olvasását és írását. A tulajdonos az ACL maszk READ_ACL és WRITE_ACL értékével vezérelheti, hogy ki olvashatja és írhatja az ACL listát. Az objektum tulajdonosa mindig READ_ATTRIBUTES, WRITE_ATTRIBUTES, READ_ACL és WRITE_ACL hozzáférsel rendelkezik. Az objektum tulajdonosa a WRITE_OWNER attribútum használatával azt is engedélyezheti mások számára, hogy módosítsák az objektum tulajdonosát és csoportját. Az objektum tulajdonosa alapértelmezésben nem tudja módosítani az objektum tulajdonosát vagy csoportját, de felvehet olyan WRITE_OWNER bejegyzést az ACL listába, amelyben saját magát adja meg, vagy az objektum örökölhet olyan ACL bejegyzést, ami WRITE_OWNER bejegyzést ad meg az OWNER@ **Ki** értékhez. A tulajdonos vagy csoport módosításakor a **setuid** bit kikapcsolásra kerül.

Néhány kivétel a szabályok alól:

- Az az objektum tulajdonosa UID 0, akkor csak UID 0 módosíthatja a tulajdonost, de a csoportot továbbra is módosíthatja a WRITE_OWNER attribútummal rendelkező alany.
- Feltételezve, hogy az objektum WRITE_OWNER attribútummal rendelkezik az alanyhoz, az 5300-05 technológiai szint előtti AIX 5.3 változatokban ha az objektum nem UID 0 tulajdonossal rendelkezik, akkor a tulajdonost csak egy másik nem UID 0 felhasználó módosíthatja. AIX with 5300-05 és újabb változatokban ha az objektum nem UID 0 tulajdonossal rendelkezik, akkor a tulajdonos csak a tulajdonost módosítani kívánó alany EUID azonosítójára módosítható.
- A csoport az alany párhuzamos csoportkészletében található bármely csoportra módosítva, annyi kivétellel, hogy sosem lehet GID 0 vagy GID 7 (rendszer vagy biztonság), még akkor sem, ha ezek a csoportok benne vannak az alany párhuzamos csoportkészletében.

Támogatott LDAP alapú és fájl alapú adminisztrációs adatbázis:

A kiértékelés nem támogatja az NFS adminisztrációs adatbázist. Az olyan hitelesítési módszerek, mint például a DCE és a NIS nem támogatottak.

A kiértékelés csak az alábbiakat támogatja:

- Fájl alapú hitelesítés (alapértelmezés)
- UNIX-stílusú LDAP-alapú hitelesítés (IBM Tivoli Directory Server v 6.0 LDAP szerver használata)

A fájl alapú hitelesítéssel kapcsolatos további információkat itt talál: Felhasználók hitelesítése.

LDAP hitelesítés:

Az LDAP alapú I&A a "UNIX típusú" hitelesítési üzemmódban állítható be. Ebben az üzemmódban az adminisztrációs adatok (beleértve a felhasználóneveket, azonosítókat és jelszavakat) az LDAP címtárban vannak, ahol az adatok elérése az LDAP adminisztrátorra korlátozott.

Amikor egy felhasználó bejelentkezik a rendszerbe, a rendszer kapcsolatot létesít az LDAP kiszolgálóval az LDAP adminisztrátori fiók használatával SSL felett, lekéri a szükséges felhasználói adatokat (beleértve a jelszót) az LDAP címtárból, majd hitelesítést végez az LDAP címtárból lekért adatok felhasználásával. A rendszer fenntart egy adminisztrációs adatbázist az LDAP kiszolgálón. A többi hoszt ugyanarról az LDAP kiszolgálóról importálja az adminisztrációs adatokat a korábban ismertetett mechanizmus használatával. A rendszer úgy tart fenn konzisztens adminisztrációs adatbázist, hogy minden adminisztrációs változtatást a kijelölt LDAP kiszolgálón hajt végre. Bármelyik számítógéphez tartozó felhasználói azonosító ugyanarra a felhasználóra vonatkozik az összes számítógépen. Továbbá a jelszó konfiguráció, a név-UID leképezések és a többi adat is megegyezik az osztott rendszer összes hosztján.

Az LDAP hitelesítés beállításával kapcsolatos további információk: Egyszerűsített címtárhozzáférési protokoll. Ha további információkra kíváncsi az SSL beállításával kapcsolatban az LDAP címtárhoz, akkor nézze meg az SSL beállítása az LDAP kiszolgálón és az SSL beállítása az LDAP ügyfélen című részt.

LDAP szerver:

Az **mksecldap -s** parancs beállít egy AIX rendszert LDAP kiszolgálóként a biztonsági hitelesítéshez és adatkezeléshez.

Tegye a következőket:

- Használja az RFC2307AIX sémát az **-S** paraméterrel.
- Állítsa be a szervert Védett socket réteg (SSL) használatára a **-k** paraméterrel. Ez a művelet 32 bites rendszereken a **GSKit V8** fájlkészlet és az **idsldap.clt_max_crypto32bit63.rte** fájlkészlet,64 bites rendszerek esetén az **idsldap.clt_max_crypto64bit63.rte** fájlkészlet telepítését igényli. Az ikeyman segédprogrammal állíthat elő kulcspárokat a címtár szerver számára.

Az LDAP felhasználó beállításait úgy kell megadni, hogy megfeleljen a kiértékelési követelményeknek. Az RFC2370AIX séma határozza meg a felhasználó attribútumait. Ugyanazokat az értékeket használja, mint a BAS/EAL4+ rendszerkonfiguráció részben leírtak. A Tivoli Directory Server adminisztrátorok nincsenek kényszerítve a jelszavuk időnkénti megváltoztatására (például nincsen **MaxAge** érték az adminisztrátori jelszavakhoz). Emiatt az LDAP adminisztrátori jelszót ugyanolyan gyakran kell módosítani, mint az AIX felhasználók jelszavát (**MaxAge** = 8 (hét)).

Tivoli Directory Server 6.3 környezetben a hitelesítési hibák kezelése nem vonatkozik a címtár adminisztrátorra vagy az adminisztrátori csoport tagjaira. A jelszóképzési szabályok nem vonatkoznak az adminisztrátori fiókokra sem. Ezeket a szabályokat be kell tartatni Tivoli Directory Server 6.3 használata esetén.

Ha az adminisztrátor nem használ közös LDAP adatbázis hátteret a felhasználókezeléshez, akkor az adminisztrátornak biztosítania kell, hogy a felhasználók hitelesítési adatait tároló adatbázis következetesen karban van tartva az egy hálózaton belüli TCP Offload Engine (TOE) rendszerrészek között. Példák:

- /etc/group
- /etc/passwd
- /etc/security/.ids
- /etc/security/.profile
- /etc/security/environ
- /etc/security/group
- /etc/security/limits
- /etc/security/passwd

- /etc/security/user

Kapcsolódó tájékoztatás:

IBM Tivoli Directory Server információk a csomagokról, fájlkészletekről és előfeltételekről

LDAP ügyfél:

Az **mksecldap -c** parancs beállít egy AIX rendszert LDAP ügyfélként a biztonsági hitelesítéshez és adatkezeléshez.

Tegye a következőket:

- Az **mksecldap -c** paranccsal adja meg a **unix_auth** értéket az **authType** beállításnak az **-A** paraméterrel.
- Állítsa be az ügyfelet az SSL használatára a **-k** paraméterrel az **mksecldap -c** parancsban. Az ügyfél SSL kulcsának megadása megköveteli a **GSKit** fájlkészlet és az **ldap.max_crypto_client** fájlkészlet telepítését. A **gsk7ikm** segédprogrammal állítsa elő a kulcspárokat a címtárkiszolgáló számára.

NFS v4 ügyfél/kiszolgáló és Kerberos:

Az NFS v4 ügyfél/kiszolgáló környezet LDAP címtárat tartalmaz a hitelesítési adatok fenntartásához és Kerberos rendszert a megbízható csatorna kialakításához az NFS v4 ügyfelek és kiszolgálók között. A kiértékelt konfiguráció támogatja a NAS v1.4 for Kerberos szoftvert és az IBM Tivoli Directory Server v6.0 változatot (LDAP szerver) a felhasználói adatbázishoz.

A NAS v1.4 (Kerberos 5-ös változatú kiszolgáló) rendszert úgy kell beállítani, hogy LDAP címtárat használjon adatbázisként. A Kerberos kiszolgáló által korábban kiadott Kerberos jegyek lejáratukig érvényesek.

Kerberos hitelesítés használatakor a felhasználó által kezdeményezett távoli eljáráshívásokban használt hitelesítési adatok a felhasználó által aktuálisan használt Kerberos jegyhez kapcsolódnak, és nem befolyásolja őket a folyamat valós vagy tényleges UID azonosítója. Amikor NFS távoli fájlrendszert ér el Kerberos hitelesítés használatával egy **setuid** program futtatása során, akkor a kiszolgálón látható UID alapja a Kerberos azonosság, nem a futtatott **setuid** programot birtokló UID.

A kiértékelt konfiguráció magában foglalja az NFS beállítását RPCSEC-GSS biztonság használatára. További információk: Hálózati fájlrendszer, NFS kiszolgáló beállítása és NFS ügyfél beállítása. A kiszolgáló beállításakor válassza a Kerberos hitelesítést és engedélyezze a kiterjesztett biztonsági szolgáltatásokat a kiszolgálón. Ezt a SMIT eszközkészleten keresztül teheti meg a **chnfs** parancs kiadásával. A **chnfs** parancs rendelkezik az RPCSEC_GSS biztonság engedélyezésére szolgáló paraméterrel. Az ügyfél beállításakor kövesse a Kerberos használatára vonatkozó útmutatást az NFS ügyfél beállítása című részből. Azzal kapcsolatos útmutatást, hogy hogyan állíthatja be a Kerberos adatkiszolgálót DES3 titkosítással a biztonság érdekében, itt talál: Hálózat beállítása RPCSEC-GSS biztonsághoz. A kiértékelt konfiguráció csak a DES3 titkosítást támogatja.

Jelszószabályok:

A kiértékelt konfigurációban a jelszószabályoknak ezekkel az értékekkel kell rendelkezniük, amikor a Kerberos kiszolgálót használja LDAP adatbázissal.

A jelszószabályokkal kapcsolatos további információkat az *IBM Network Authentication Service Version 1.4 for AIX, Linux and Solaris adminisztrátori és felhasználói kézikönyv* "9. fejezet: Hálózati hitelesítési szolgáltatás jelszók kezelése" című része tartalmaz.

Az értékek listája:

mindiff
4

maxrepeats
2

minalpha
2

minother
2

minlen 8

minage
0

histsize
10

Ahhoz, hogy az AIX NFS v4 ügyfél és az AIX NFS v4 kiszolgáló biztonságosan kommunikáljon kifejezetten csak DES3 titkosítási típusok használatával, hozza létre az enctypes, "nfs/hostname" kiszolgálóazonosítót DES3 titkosítási típussal (például des3-cbc-sha1), valamint a megfelelő bejegyzést a keytab fájlban (a **kadmin** felület használatával), és legyen a DES3 (például **des3-cbc-sha1**) az első bejegyzés az NFS v4 számítógép /etc/krb5/krb5.conf fájljának **default_tgs_enctypes** szakaszában.

Virtuális I/O szerver:

A Virtuális I/O szerver (VIOS) egy különálló LPAR partíción található, és leképezéseken keresztül alapszintű tetszés szerinti hozzáférés-felügyeletet biztosít az LPAR partíciók nevében tevékenykedő VIOS SCSI eszközillesztők és az SCSI alapú logikai kötetek és fizikai kötetek között.

Egy LPAR partíció (egy VIOS SCSI eszközillesztőn keresztül) 0 vagy több logikai és fizikai kötetre képezhető le, de egy kötet csak egy LPAR partícióra képezhető le. Ez a leképezés csak a hozzárendelt kötetekre korlátozza az LPAR partíciót. A VIOS vezérli a VIOS Ethernet adapter eszközillesztők leképezését is a virtuális hálózaton osztozó LPAR partíciók csoportjainak nevében tevékenykedő VIOS Ethernet eszközillesztőkre. A kiértékelt konfigurációban csak egy az egyhez leképezés engedélyezett az Ethernet adapter eszközillesztőkről az LPAR partíciók csoportja nevében tevékenykedő Ethernet eszközillesztőkre. Az egy az egyhez leképezést az adminisztrátor állítja be és az eszközillesztők tartatják be. Valamint az Ethernet csomagokat nem lehet VLAN címkével ellátni a kiértékelt konfigurációban. Ezzel a mechanizmussal korlátozható, hogy mely LPAR partíciók látnak bizonyos Ethernet csomagokat.

A VIOS csatolót védeni kell a jogosulatlan felhasználók általi eléréstől. Az VIOS felhasználó beállításait úgy kell megadni, hogy megfeleljen a kiértékelési követelményeknek. A tényleges követelmény az, hogy a TSF-nek mechanizmust kell biztosítania annak ellenőrzésére, hogy a titkok megfelelnek az alábbi minőségi mérésnek: annak valószínűsége, hogy hogy egy támadó meg tud szerezni egy titkot az adott titok élettartama során, kisebb, mint 2^-20. Az alábbi paramétereket kell módosítani a felhasználóhoz az /etc/security/user könyvtárban:

maxage
8

maxexpired
1

minother
2

minlen 8

maxrepeats
2

loginretries
3

histexpire
52

histsize
20

Az alapértelmezett értékek módosításához adja ki az alábbi parancsokat:

```
type oem_setup_env

chsec -f /etc/security/user -s default  -a maxage=8 -a maxexpired=1 -a minother=2
  -a minlen=8 -a maxrepeats=2 -a loginretries=3 -a histexpire=52 -a histsize=20
```

Amikor az elsődleges adminisztrátor (**padmin**) új felhasználót hoz létre, akkor a felhasználói attribútumokat kifejezetten meg kell adni a felhasználóhoz. Ha például létre akar hozni egy új felhasználót *davis* néven, akkor a **padmin** az alábbi parancsot használja:

```
mkuser  maxage=8 maxexpired=1 minother=2 minlen=8 maxrepeats=2 loginretries=3
  histexpire=52 histsize=20 davis
```

A **padmin** feladata továbbá az alábbi démonok leállítása, majd a rendszer újraindítása:

- A **writesrv** és a **ctrmc** eltávolítása az /etc/inittab fájlból:

  ```
  sshd:   stopsrc -s sshd
  ```

- Annak megakadályozásához, hogy a démon elinduljon a rendszerbetöltéskor, távolítsa el az /etc/rc.d/rc2.d/Ksshd és az /etc/rc.d/rc2.d/Ssshd fájlt. Az újraindítás után állítsa le az RSCT démonokat:

  ```
  stopsrc -g rsct_rm stopsrc -g rsct
  ```

Minden felhasználó, szerepétől függetlenül, adminisztrátori felhasználónak számít.

A rendszeradminisztrátor minden parancsot futtathat, kivéve az alábbi listán szereplőeket, amelyek az elsődleges adminisztrátorra (**padmin**) korlátozottak:

- **chdate**
- **chuser**
- **cleargcl**
- **de_access**
- **diagmenu**
- **invscout**
- **loginmsg**
- **lsfailedlogin**
- **lsgcl**
- **mirrorios**
- **mkuser**
- **motd**
- **oem_platform_level**
- **oem_setup_env**
- **redefvg**
- **rmuser**
- **leállítás**
- **unmirrorios**

Bejelentkezés felügyelete

A rendszer telepítése után biztonsági okokból megváltoztathatja a bejelentkezési képernyő alapértelmezéseit.

A potenciális betörők értékes információkat szerezhetnek az alapértelmezett AIX bejelentkezési képernyőből, például a hoszt nevét és az operációs rendszer verziószámát. Ezen információk alapján sokkal könnyebben meghatározhatják

azokat a behatolási módszereket, amelyekkel érdemes megpróbálkozni. Biztonsági okokból érdemes a bejelentkezési képernyő alapértelmezéseit a rendszertelepítés után a lehető leghamarabb lecserélni.

A KDE és GNOME munkaasztalok esetében is felmerül néhány biztonsági kérdés. A KDE és GNOME felületekről további információkat az *Installation and migration* című kiadványban olvashat.

A felhasználókat, csoportokat és jelszavakat a “Felhasználók, csoportok és jelszavak” oldalszám: 46 rész tárgyalja részletesen.

Bejelentkezési vezérlők beállítása:

Bejelentkezési vezérlők beállítása a /etc/security/login.cfg fájlban.

A jelszókitalálásos rendszertámadás megnehezítése érdekében /etc/security/login.cfg fájlban állítsa be a következő bejelentkezési vezérlőket az:

1. táblázat: Bejelentkezés felügyelethez kapcsolódó attribútumok és ajánlott értékek

Attribútum	PTY (hálózati) eszközökre vonatkozik	TTY eszközökre vonatkozik	Ajánlott érték	Megjegyzések
sak_enabled	I	I	false	A biztonságos figyelmeztetés ritkán szükséges. Lásd: “Biztonságos figyelem billentyű használata” oldalszám: 5.
logintimes	N	I		Itt adhatja meg a megengedett bejelentkezési időszakokat.
logindisable	N	I	4	Bejelentkezés letiltása a terminálon 4 egymást követő sikertelen kísérlet után.
logininterval	N	I	60	A terminál akkor kerül letiltásra, ha a megadott érvénytelen kísérletek 60 másodpercen belül történnek.
loginreenable	N	I	30	A terminál ismételt engedélyezése 30 percnyi tiltás után.
logindelay	I	I	5	A bejelentkezési felszólítások közötti idő másodpercben mérve. Az értéket a rendszer beszorozza a sikertelen kísérletek számával, így sikertelen bejelentkezések esetén 5, 10, 15 és 20 másodperces várakozással kell számolni.

Ezek a port korlátozások leginkább a soros csatlakozású terminálokon működnek, a hálózati bejelentkezésekhez használt pszeudoterminálokon nem. A fájlban a terminálok kifejezetten is megadhatók, például:

```
/dev/tty0:
        logintimes = 0600-2200
        logindisable = 5
        logininterval = 80
        loginreenable = 20
```

Bejelentkezési képernyő üdvözlő üzenetének módosítása:

Ha meg kívánja akadályozni bizonyos információk megjelenését a bejelentkezési képernyőn, akkor módosítsa az /etc/security/login.cfg fájl *herald* paraméterét.

Az alapértelmezett *herald* a bejelentkezési képernyőn megjelenő üzenetet tartalmazza. A paraméter módosításához használja a **chsec** parancsot, vagy módosítsa közvetlenül a fájlt.

Az alábbi példa a **chsec** parancs segítségével módosítja az alapértelmezett *herald* paramétert:

```
# chsec -f /etc/security/login.cfg -s default
-a herald="Unauthorized use of this system is prohibited.\n\nlogin:"
```

A **chsec** parancsról további információkat az *Commands Reference, Volume 1* című kiadványból szerezhet.

A fájl közvetlen szerkesztéséhez nyissa meg a /etc/security/login.cfg fájlt és frissítse a *herald* paramétert az alábbi módon:

```
default:
herald ="Unauthorized use of this system is prohibited\n\nlogin:"
 sak_enable = false
 logintimes =
 logindisable = 0
 logininterval = 0
 loginreenable = 0
 logindelay = 0
```

Megjegyzés: A rendszer biztonságosabbá tételéhez állítsa be a *logindisable* és *logindelay* változó értékét 0-nál nagyobb számra (# > 0).

Egységes munkaasztali környezet bejelentkezési képernyőjének módosítása:

Ez a biztonsági kérdés az Common Desktop Environment (CDE) felhasználókat is érinti. A CDE bejelentkezési képernyő alapértelmezésben szintén megjeleníti a hosztnevet és az operációs rendszer verziószámát. Ezek megjelenítésének letiltásához módosítsa az /usr/dt/config/$LANG/Xresources fájlt, ahol a **$LANG** a számítógépre telepített helyi nyelvnek felel meg.

A példában, feltételezve, hogy a **$LANG** értéke **C**, másolja át ezt a fájlt a /etc/dt/config/C/Xresources könyvtárba. Ezután nyissa meg az /usr/dt/config/C/Xresources fájlt, és távolítsa el belőle a hosztnevet és operációs rendszer verziószámot tartalmazó üdvözlő üzeneteket.

A CDE biztonsági kérdéseiről további információkat az “X11 és CDE problémák kezelése” oldalszám: 39 szakaszban olvashat.

A felhasználónév kijelzésének letiltása és a jelszóhoz tartozó üzenet módosítása:

Biztonságos környezetben bejelentkezéskor szükség lehet a felhasználónév elrejtésére, vagy az alapértelmezettől eltérő bejelentkezési jelszóhoz tartozó üzenet megjelenítésére.

A bejelentkezés és jelszóhoz tartozó üzenet alapértelmezett viselkedését az alábbi példa mutatja:

```
login: foo
foo's Password:
```

Ha le kívánja tiltani a felhasználónév kijelzését a bejelentkező képernyőn és a hibaüzenetekben, akkor módosítsa a *usernameecho* paramétert az /etc/security/login.cfg fájlban. A *usernameecho* értéke alapértelmezésben true, ami a felhasználónév megjelenítését jelenti. A paraméter módosításához használja a **chsec** parancsot, vagy módosítsa közvetlenül a fájlt.

Az alábbi példa a **chsec** parancs segítségével változtatja meg a *usernameecho* paraméter értékét false-ra:

```
# chsec -f /etc/security/login.cfg -s default -a usernameecho=false
```

A **chsec** parancsról további információkat az *Commands Reference, Volume 1* című kiadványból szerezhet.

A fájl közvetlen szerkesztéséhez nyissa meg az /etc/security/login.cfg fájlt, és módosítsa vagy hozza létre a *usernameecho* bejegyzést az alábbiak szerint:

```
default:
 usernamecho = false
```

Ha a *usernameecho* paramétert false értékre állítja, akkor a bejelentkező képernyőn illetve a hibaüzenetekben a felhasználónév helyett csillag (*) karakterek jelennek meg az alábbiakhoz hasonló módon:

```
login:
***'s Password:
```

A jelszóhoz tartozó üzenet külön módosítható, hogy egyedi karaktersorozat legyen az /etc/security/login.cfg fájlban található *pwdprompt* paraméter beállításával. Az alapértelmezett karaktersorozat: "*felhasználó*'s Password: ", ahol a *felhasználó* változó helyére az éppen bejelentkező felhasználó neve helyettesítődik be:

A paraméter módosításához használja a **chsec** parancsot, vagy módosítsa közvetlenül a fájlt.

Az alábbi példa a **chsec** parancs segítségével módosítja az alapértelmezett *pwdprompt* paramétert "Jelszó: "-ra:

```
# chsec -f /etc/security/login.cfg -s default -a pwdprompt="Jelszó: "
```

A fájl közvetlen szerkesztéséhez nyissa meg az /etc/security/login.cfg fájlt, és módosítsa vagy hozza létre a *pwdprompt* bejegyzést az alábbiak szerint:

```
default:
 pwdprompt = "Jelszó: "
```

Ha a *pwdprompt* paraméterhez a "Jelszó: " értéket rendeli, akkor a bejelentkezéskor, és más, a rendszer bejelentkező képernyőjét használó alkalmazások indításakor a megadott üzenet fog megjelenni. A fenti beállítások hatására a bejelentkező képernyő viselkedése az alábbiak szerint alakul:

```
login: foo
Jelszó:
```

A rendszer alapértelmezett bejelentkezési paramétereinek beállítása:

Módosítsa az /etc/security/login.cfg fájlt rendszer alapértelmezett bejelentkezési paraméterek beállításához.

Az /etc/security/login.cfg fájl módosításával állítson be olyan alapértelmezéseket (bejelentkezési ismétlések száma, bejelentkezés ismételt engedélyezése, bejelentkezési időszak), amelyeket az új felhasználóknak beállítana.

Felügyelet nélküli terminálok védelme:

A **lock** és **xlock** parancs segítségével védje a terminált.

Minden rendszer sérülékeny, ha bejelentkezett, de felügyelet nélkül hagyott terminálok csatlakoznak rá. A legsúlyosabb problémát az jelenti, amikor az adminisztrátor otthagy egy root felhasználóval bejelentkezett terminált. A felhasználóknak általában minden esetben ki kell lépniük, ha elhagyják terminálju kat. A terminálok felügyelet nélkül hagyása súlyos biztonsági kockázatot jelent. A terminál zárolásához használja a **lock** parancsot. AIXwindows felület esetén használja az **xlock** parancsot.

Automatikus kijelentkezés engedélyezése:

Az automatikus kijelentkezés engedélyezésével megakadályozható, hogy behatoló veszélyeztesse a rendszer biztonságát.

Szintén biztonsági problémát jelentenek az olyan felhasználók, akik hosszabb ideig felügyelet nélkül hagyják felhasználói fiókjukat. Ez a helyzet lehetővé teszi a behatolóknak, hogy átvegyék a felhasználó terminálját, és veszélyeztessék a rendszer biztonságát.

Az ilyen jellegű biztonsági kockázatok kivédése érdekében engedélyezheti a rendszeren az automatikus kijelentkezést. Ehhez a TMOUT és a TIMEOUT környezeti változóban állítsa be az inaktivitási időkorlátot másodpercben. Az inaktív idő letelte után a rendszer automatikusan kijelentkezteti a felhasználót. Például:

```
TMOUT=600; TIMEOUT=600; export TMOUT TIMEOUT
```

A fenti példában az időkorlát 600 másodperc, vagyis 10 perc. Ez a módszer kizárólag a héj alkalmazásból működik. Ha meg akarja védeni a változókat a véletlen felülírástól, akkor tegye őket csak olvashatóvá, az alábbiak szerint:

```
readonly TMOUT TIMEOUT
```

A TMOUT és TIMEOUT környezeti változók a felhasználók .profile fájljában vagy az /etc/security/.profile fájlban vannak beállítva. Ez lehetővé teszi a fájl hozzáadását a felhasználó .profile fájljához a felhasználó létrehozásakor.

Veremvégrehajtás letiltása védelem

A számítógéprendszer biztonságos formában tartása az Igény szerinti (On Demand) kereskedelem fontos szempontja. A mai világ hálózatos környezeteiben nagy kihívást jelent a különböző forrásokból érkező támadások kivédése.

A számítógéprendszerek egyre nagyobb valószínűséggel esnek áldozatul kifinomult támadásoknak, amelyek a vállalatok és kormányzati ügynökségek napi működését akadályozhatják. Amíg nincsenek olyan biztonsági intézkedések, amelyek teljesen biztos védelmet biztosítanak a támadások ellen, addig több biztonsági mechanizmust kell telepíteni a biztonsági támadások megakadályozása érdekében. Ez a fejezet az AIX által a puffertúlcsordulás miatt fellépő kivételt kihasználó támadások kivédése ellen használt biztonsági mechanizmusokat mutatja be.

Többféle biztonsági rés lehetséges, de az egyik legáltalánosabb metódus a rendszer által biztosított adminisztrációs eszközök megfigyelése, a puffertúlcsordulások keresése és kihasználása. Puffertúlcsordulás támadás akkor történhet, ha egy belső program felülírásra kerül, mivel az adatok nem kerültek megfelelően ellenőrzésre (például a parancssor, környezeti változó, lemez vagy terminál I/O). A támadókód a puffertúlcsordulás során bekerül a futó folyamatba, és megváltoztatja a futó folyamat végrehajtási útját. A visszatérési cím felülírásra kerül és átirányítódik a beszúrt kód helyére. A rések általános oka a helytelen vagy nem létező határellenőrzés, vagy az adatforrások érvényességének helytelen feltételezései. Puffertúlcsordulás például akkor történhet, ha az adatobjektum elég nagy ahhoz, hogy 1 KB adatot tároljon, de a program nem ellenőrzi a bemenet határait, ezáltal 1 KB-nál több adat is másolható az objektumba.

A behatoló célja, hogy megtámadjon egy olyan parancsot és/vagy eszközt, amely root jogosultságokat biztosít egy normál felhasználó számára. A program vezérlése az összes jogosultsággal kerül átadásra, így lehetővé teszi a pufferek túlcsordulását. A támadások jellemzően a root birtokában lévő UID készletet vagy halmazokat célozzák meg, amelyek egy parancsfájl végrehajtásához vezetnek, ezáltal root alapú parancsértelmező hozzáférést biztosítanak a rendszerhez.

Ezek a támadások a puffertúlcsordulás során beírt támadókód végrehajtásának blokkolásával akadályozhatók meg. Tiltsa le a végrehajtást a folyamat azon memóriaterületein, ahol általában nem történik végrehajtás (verem és kupac memóriaterületek).

SED puffertúlcsordulás védelmi mechanizmus:

AIX engedélyezte a veremvégrehajtás letiltása (SED) mechanizmust a kód végrehajtásának letiltása érdekében a vermen és a folyamat adatterületeinek kiválasztása érdekében.

Egy sérült program végrehajtásának letiltásával majd leállításával megakadályozható, hogy a támadó root jogosultságokat szerezzen puffertúlcsordulás esetén. Ez a szolgáltatás nem állítja meg a puffertúlcsordulást, de védelmet biztosít azáltal, hogy letiltja a túlcsordult pufferen a támadások végrehajtását.

A POWER4 processzorcsaládtól kezdve használható egy oldal-szintű végrehajtás engedélyezési és/vagy letiltási funkció a memóriához. A AIX SED mechanizmus ezt az alapul szolgáló hardvertámogatást használja egy nem végrehajtási funkció megvalósításához a kiválasztott memóriaterületeken. Ha ez a szolgáltatás engedélyezve van, akkor az operációs rendszer ellenőrzi és jelzi a különböző fájlokat a végrehajtható programok során. Ezután riasztja az operációs rendszer memóriakezelőt és a folyamatkezelőket, hogy a SED engedélyezve van a létrehozandó folyamathoz. A kiválasztott memóriaterületek "nem végrehajtás" jelzést kapna. Ha végrehajtás történik ezeken a megjelölt területeken, akkor a hardver bebillent egy végrehajtási jelzőt és az operációs rendszer leállítja a megfelelő folyamatot. A végrehajtás és az alkalmazás-végrehajtás részletei az AIX hibanapló-eseményeken keresztül érhetők el.

A SED megvalósítása főként a **sedmgr** parancson keresztül történik. A **sedmgr** parancs lehetővé teszi a működés rendszerszintű SED mód vezérlését valamint a végrehajtható fájl beállítását a SED jelzők alapján.

SED módok és megfigyelés:

Az AIX veremvégrehajtás letiltás (SED) mechanizmusa rendszerszintű módjelzők valamint egyedi végrehajtható fájl alapú fejlécjelzők segítségével kerül megvalósításra.

A rendszerszintű jelzők a SED rendszerszintű működését szabályozzák, a fájlszintű jelzők pedig jelzik, hogy a fájlokat a SED-ben hogyan kell kezelni. A puffertúlcsordulás védelmi (BOP) mechanizmus négy rendszerszintű működési módot biztosít:

off A SED mechanizmus kikapcsolásra kerül és egy eljárás sem lesz védve SED-del.

select A fájlok csak egy kijelölt része kerül engedélyezésre és megfigyelésre SED védelemhez. A fájlok kijelölt része a végrehajtható program bináris fejlécében lévő SED-del kapcsolatos jelzők áttekintésével kerül kiválasztásra. A végrehajtható program fejléc lehetővé teszi, hogy a SED-del kapcsolatos jelzők bekerülhessenek **select** módba.

setidfiles
Lehetővé teszi a SED engedélyezését nem csak az ilyen mechanizmust kérő fájlokhoz, hanem az összes fontos **setuid** és **setgid** rendszerfájlhoz is. Ebben a módban az operációs rendszer nem csak a beállított **request** SED jelzővel rendelkező fájlokhoz biztosít SED-et, hanem a következő karakterisztikával rendelkező végrehajtható fájlokhoz is engedélyezi a SED-et (azon fájlok kivételével, amelyek fájlfejlécében *exempt* látható):

- root tulajdonában lévő SETUID fájlok
- SETGID fájlok, amelyek elsődleges csoportja **system** vagy **security**

all A rendszer minden betöltött végrehajtható fájlja SED-del védett, kivéve azokat, amelyek mentességet kérnek a SED mód alól. A mentességgel kapcsolatos jelzők a végrehajtható programfejlécek részei.

Az AIX SED funkciója a folyamat leállítása helyett megfigyelést is lehetővé tesz kivétel fellépése esetén. Ezen rendszerszintű vezérlés segítségével a rendszeradminisztrátorok a rendszer megfigyelésével ellenőrizhetik a rendszerkörnyezet összeomlásait és problémáit, a SED éles rendszereken telepítése előtt.

A **sedmgr** parancs egy beállítást biztosít, amelynek segítségével a SED kivételek fellépése esetén a folyamatokat leállítás helyett megfigyelheti. A rendszeradminisztrátor kiértékelheti, hogy a végrehajtható program szabályszerű veremvégrehajtást végez-e. Ez a beállítás a -c kapcsolóval megadott rendszerszintű móddal együtt működik. Ha a **monitor** mód be van kapcsolva, akkor a rendszer lehetővé teszi a folyamat számára a működés folytatását SED-del kapcsolatos kivétel fellépése esetén. A folyamat leállítása helyett az operációs rendszer naplózza a kivételt az AIX hibanaplóban. Ha a SED megfigyelés ki van kapcsolva, akkor az operációs rendszer leállít minden folyamatot, amely kivételt okoz SED szolgáltatásonként.

A SED mód rendszerszintű jelzőinek módosításakor a rendszer újra kell indítani, hogy a módosítások érvényre jussanak. Az ilyen típusú események megfigyelésre kerülnek.

SED jelzők a végrehajtható fájlokhoz:

AIX rendszerben a **sedmgr** parancs segítségével jelezheti a végrehajtható fájlokat az SE mechanizmusból.

Az összekapcsoló (linker) kibővítésre került két új SED-hez kapcsolódó jelző támogatására, hogy a végrehajtható állományok fejlécében lehetőséget adjon a select és az exempt beállításokra. A **select** jelző módot ad egy végrehajtható állomány számára a SED védelem kérésére és ennek részévé válásra a rendszerszintű SED műveletek select módja során, míg az **exempt** jelző lehetőséget ad arra, hogy egy végrehajtható állomány mentességet kérjen a SED mechanizmusoktól. Ezek a végrehajtható fájlok nincsenek engedélyezve folyamatmemória-területek végrehajtásának letiltásához.

A mentesség jelző segítségével a rendszeradminisztrátor megfigyelheti a SED mechanizmust és kiértékelheti a helyzetet. A rendszeradminisztrátor engedélyezheti a végrehajtást a vermen és adatterületeken, ahogy az alkalmazáshoz szükséges, a hozzátartozó kockázatok megértésével.

A következő táblázat bemutatja, hogy a rendszerszintű beállítások és a fájlbeállítások hogyan hatnak a működés SED módjára:

2. táblázat: A SED módot befolyásoló rendszerszintű beállítások és fájlbeállítások

Rendszer SED mód	Végrehajtható fájl SED jelzők			Setuid-root vagy setgid-system/security fájlok
	kérés	mentesség	rendszer	
off	–	–	–	–
select	engedélyezett	–	–	–
setgidfiles	engedélyezett	–	–	engedélyezett
all	engedélyezett	–	engedélyezett	engedélyezett

SED problémák és szempontok:

Alapértelmezés szerint az AIX SED **select** módot használ. Egy sor **setuid** és **setgid** program a **select** móddal kiválasztható a SED számára, és alapértelmezésben védett módban működik.

A SED-engedélyezés következtében a régebbi bináris állományok megszakadhatnak, ha nem képesek a veremkupac-területek no-execution funkciójának végrehajtására. Ezeknek az alkalmazásoknak veremadat-területeken kell futnia. A rendszeradminisztrátor kiértékelheti a helyzetet és jelölhetik a fájlt mentességre a **bopmgr** parancs segítségével. AIX Java™ 1.3.1 és AIX Java 1.4.2 Just-In-Time (JIT) fordítókkal rendelkezik, amelyek dinamikusan állítanak elő és hajtanak végre eredeti objektumkódokat a Java alkalmazások futtatása során (a Java Virtual Machine eldönti, hogy mely kódot kell végrehajtani az alkalmazás végrehajtási profilja alapján). Ez az objektumkód a JIT által lefoglalt adatpufferekben kerül tárolásra. Következésképp ha a AIX úgy van beállítva, hogy a SED **ALL** módban fusson, akkor a rendszeradminisztrátornak be kell állítania a Java bináris fájl mentességi jelzőjét.

Ha egy végrehajtható fájl SED-fel kapcsolatos jelzői módosításra kerülnek, akkor csak a jövőbeli fájlbetöltésre és -végrehajtásra kerülnek alkalmazásra. Ez a módosítás nem érvényes a jelenleg működő folyamatokra a fájlon. A SED szolgáltatás vezérli és megfigyeli a 32 és 64 bites végrehajtható programok rendszerszintű és fájlszintű beállításait. A SED szolgáltatás csak akkor áll rendelkezésre, ha az AIX operációs rendszert 64 bites kernellel használja.

Kapcsolódó információk

sedmgr command

AIX **Hibanaplózási szolgáltatás**

X11 és CDE problémák kezelése

Az X11 X szerverrel és az Egységes munkaasztali környezettel (CDE) kapcsolatban biztonsági veszélyek merülhetnek fel.

A /etc/rc.dt fájl eltávolítása:

magas szintű biztonsági követelményeket támasztó szervereken távolítsa el a /etc/rc.dt fájlt.

Bár a CDE felület futtatása kényelmes a felhasználók számára, használatuk kapcsán felmerül néhány biztonsági probléma is. Ezen okból ne futtassa a CDE-t magas szintű biztonsági követelményeket támasztó szervereken. A legjobb megoldás a CDE (dt) fájlkészletek telepítésének elkerülése. Ha telepítette ezeket a fájlkészleteket a rendszeren, akkor fontolja meg eltávolításukat, különösen a CDE indítását végző /etc/rc.dt fájlét.

A CDE-ről további információkat az *Operating system and device management* című kiadványban talál.

Távoli X szerver jogosulatlan megfigyelésének megakadályozása:

Az X11 szerverekkel kapcsolatos egyik fontos biztonsági probléma a távoli szerverek jogosulatlan csendes megfigyelése.

Az **xwd** és **xwud** parancsok használhatók az X szerver tevékenységének figyelésére, mivel képesek a billentyűleütések elfogására, amely jelszavak és más érzékeny adatok megjelenítéséhez vezethetnek. A probléma megoldásához távolítsa el ezeket a futtatható fájlokat, vagy ha ez nem megoldható, akkor korlátozza elérésüket a root felhasználóra.

Az **xwd** és **xwud** parancs a X11.apps.clients fájlkészletben található.

Ha nem szükséges az **xwd** és **xwud** parancsok megtartása, akkor fontolja meg az OpenSSH vagy az MIT Magic Cookies használatát. Ezek a harmadik féltől származó alkalmazások segítenek elkerülni az **xwd** és **xwud** parancs futtatásából adódó kockázatokat.

Az OpenSSH és az MIT Magic Cookies programokkal kapcsolatos további információkat a dokumentációban találhatók.

Hozzáférés felügyelet engedélyezése és tiltása:

Az X szerver lehetővé teszi a távoli hosztoknak, hogy az **xhost +** paranccsal csatlakozzanak a rendszerre.

Az **xhost +** paranccsal adjon meg egy hosztnevet, mivel ez letiltja az X szerver hozzáférés felügyeletét. Ez lehetővé teszi, hogy bizonyos hosztoknak engedélyezze a hozzáférést; ez megkönnyíti az X szerver potenciális támadásainak megfigyelését. Ha egy adott hosztnak hozzáférést kíván biztosítani, akkor futtassa az **xhost** parancsot az alábbiak szerint:

```
# xhost + hosztnév
```

Ha nem ad meg hosztnevet, akkor minden hoszt hozzáférést nyer.

Az **xhost** paranccsal kapcsolatos további információkat az alábbi részben talál: *Commands Reference*

Az xhost parancs futtatására szolgáló felhasználói jogosultságok letiltása.:

Az **xhost** parancs jogosulatlan végrehajtása a **chmod** parancs segítségével megakadályozható.

Az **xhost** parancs helyénvaló használatának biztosítására egy másik módszer, hogy a parancs futtatását csak root felhasználóknak engedélyezi. Ehhez a **chmod** paranccsal módosítsa az /usr/bin/X11/xhost jogosultságait 744-re az alábbiak szerint:

```
chmod 744 /usr/bin/X11/xhost
```

setuid/setgid programok listája

Különböző setuid/setgid programok állnak rendelkezésre az AIX rendszeren. Ezek a jogosultságok eltávolíthatók azokról a parancsokról, amelyeknek a normál felhasználók számára nem kell rendelkezésre állniuk.

A normál AIX telepítés a következő programokat tartalmazza. CC beállítású AIX rendszeren ez a lista kevesebb programot tartalmaz.

- /opt/IBMinvscout/bin/invscoutClient_VPD_Survey
- /opt/IBMinvscout/bin/invscoutClient_PartitionID
- /usr/lpp/diagnostics/bin/diagsetrto
- /usr/lpp/diagnostics/bin/Dctrl
- /usr/lpp/diagnostics/bin/diagela
- /usr/lpp/diagnostics/bin/diagela_exec
- /usr/lpp/diagnostics/bin/diagrpt

- /usr/lpp/diagnostics/bin/diagrto
- /usr/lpp/diagnostics/bin/diaggetrto
- /usr/lpp/diagnostics/bin/update_manage_flash
- /usr/lpp/diagnostics/bin/utape
- /usr/lpp/diagnostics/bin/uspchrp
- /usr/lpp/diagnostics/bin/update_flash
- /usr/lpp/diagnostics/bin/uesensor
- /usr/lpp/diagnostics/bin/usysident
- /usr/lpp/diagnostics/bin/usysfault
- /usr/lpp/X11/bin/xlock
- /usr/lpp/X11/bin/aixterm
- /usr/lpp/X11/bin/xterm
- /usr/lpp/X11/bin/msmitpasswd
- /usr/lib/boot/tftp
- /usr/lib/lpd/digest
- /usr/lib/lpd/rembak
- /usr/lib/lpd/pio/etc/piodmgrsu
- /usr/lib/lpd/pio/etc/piomkpq
- /usr/lib/lpd/pio/etc/pioout
- /usr/lib/mh/slocal
- /usr/lib/perf/libperfstat_updt_dictionary
- /usr/lib/sa/sadc
- /usr/lib/semutil
- /usr/lib/trcload
- /usr/sbin/allocp
- /usr/sbin/audit
- /usr/sbin/auditbin
- /usr/sbin/auditcat
- /usr/sbin/auditconv
- /usr/sbin/auditmerge
- /usr/sbin/auditpr
- /usr/sbin/auditselect
- /usr/sbin/auditstream
- /usr/sbin/backbyinode
- /usr/sbin/cfgmgr
- /usr/sbin/chcod
- /usr/sbin/chcons
- /usr/sbin/chdev
- /usr/sbin/chpath
- /usr/sbin/chtcb
- /usr/sbin/cron
- /usr/sbin/acct/accton
- /usr/sbin/arp64
- /usr/sbin/arp
- /usr/sbin/devinstall

- /usr/sbin/diag_exec
- /usr/sbin/entstat
- /usr/sbin/entstat.ethchan
- /usr/sbin/entstat.scent
- /usr/sbin/diskusg
- /usr/sbin/exec_shutdown
- /usr/sbin/fdformat
- /usr/sbin/format
- /usr/sbin/fuser
- /usr/sbin/fuser64
- /usr/sbin/getlvcb
- /usr/sbin/getlvname
- /usr/sbin/getvgname
- /usr/sbin/grpck
- /usr/sbin/getty
- /usr/sbin/extendvg
- /usr/sbin/fastboot
- /usr/sbin/frcactrl64
- /usr/sbin/frcactrl
- /usr/sbin/inetd
- /usr/sbin/invscout
- /usr/sbin/invscoutd
- /usr/sbin/ipl_varyon
- /usr/sbin/keyenvoy
- /usr/sbin/krlogind
- /usr/sbin/krshd
- /usr/sbin/lchangelv
- /usr/sbin/lchangepv
- /usr/sbin/lchangevg
- /usr/sbin/lchlvcopy
- /usr/sbin/lcreatelv
- /usr/sbin/ldeletelv
- /usr/sbin/ldeletepv
- /usr/sbin/lextendlv
- /usr/sbin/lmigratelv
- /usr/sbin/lmigratepp
- /usr/sbin/lparsetres
- /usr/sbin/lpd
- /usr/sbin/lquerylv
- /usr/sbin/lquerypv
- /usr/sbin/lqueryvg
- /usr/sbin/lqueryvgs
- /usr/sbin/lreducelv
- /usr/sbin/lresynclp
- /usr/sbin/lresynclv

- /usr/sbin/lsaudit
- /usr/sbin/lscfg
- /usr/sbin/lscons
- /usr/sbin/lslv
- /usr/sbin/lspath
- /usr/sbin/lspv
- /usr/sbin/lsresource
- /usr/sbin/lsrset
- /usr/sbin/lsslot
- /usr/sbin/lsuser
- /usr/sbin/lsvg
- /usr/sbin/lsvgfs
- /usr/sbin/login
- /usr/sbin/lvaryoffvg
- /usr/sbin/lvaryonvg
- /usr/sbin/lvgenmajor
- /usr/sbin/lvgenminor
- /usr/sbin/lvrelmajor
- /usr/sbin/lvrelminor
- /usr/sbin/lsmcode
- /usr/sbin/mailq
- /usr/sbin/mkdev
- /usr/sbin/mklvcopy
- /usr/sbin/mknod
- /usr/sbin/mkpasswd
- /usr/sbin/mkpath
- /usr/sbin/mkvg
- /usr/sbin/mount
- /usr/sbin/netstat64
- /usr/sbin/mtrace
- /usr/sbin/ndp
- /usr/sbin/newaliases
- /usr/sbin/named9
- /usr/sbin/named8
- /usr/sbin/netstat
- /usr/sbin/nfsstat
- /usr/sbin/pdelay
- /usr/sbin/pdisable
- /usr/sbin/penable
- /usr/sbin/perf/diag_tool/getschedparms
- /usr/sbin/perf/diag_tool/getvmparms
- /usr/sbin/phold
- /usr/sbin/portmir
- /usr/sbin/pshare
- /usr/sbin/pstart

- /usr/sbin/putlvcb
- /usr/sbin/putlvodm
- /usr/sbin/qdaemon
- /usr/sbin/quota
- /usr/sbin/reboot
- /usr/sbin/redefinevg
- /usr/sbin/repquota
- /usr/sbin/restbyinode
- /usr/sbin/rmdev
- /usr/sbin/ping
- /usr/sbin/rmgroup
- /usr/sbin/rmpath
- /usr/sbin/rmrole
- /usr/sbin/rmuser
- /opt/rsct/bin/ctstrtcasd
- /usr/sbin/srcd
- /usr/sbin/srcmstr
- /usr/sbin/rmsock64
- /usr/sbin/sendmail_ssl
- /usr/sbin/sendmail_nonssl
- /usr/sbin/rmsock
- /usr/sbin/sliplogin
- /usr/sbin/sendmail
- /usr/sbin/rwhod
- /usr/sbin/route
- /usr/sbin/snappd
- /usr/sbin/swap
- /usr/sbin/swapoff
- /usr/sbin/swapon
- /usr/sbin/swcons
- /usr/sbin/switch.prt
- /usr/sbin/synclvodm
- /usr/sbin/tsm
- /usr/sbin/umount
- /usr/sbin/umountall
- /usr/sbin/unmount
- /usr/sbin/varyonvg
- /usr/sbin/watch
- /usr/sbin/talkd
- /usr/sbin/timedc
- /usr/sbin/uucpd
- /usr/bin/bellmail
- /usr/bin/at
- /usr/bin/capture
- /usr/bin/chcore

- /usr/bin/acctras
- /usr/bin/acctctl
- /usr/bin/chgroup
- /usr/bin/chkey
- /usr/bin/chque
- /usr/bin/chquedev
- /usr/bin/chrole
- /usr/bin/chsec
- /usr/bin/chuser
- /usr/bin/confsrc
- /usr/bin/crontab
- /usr/bin/enq
- /usr/bin/filemon
- /usr/bin/errpt
- /usr/bin/fileplace
- /usr/bin/fileplacej2
- /usr/bin/fileplacej2_64
- /usr/bin/ftp
- /usr/bin/getconf
- /usr/bin/ipcs
- /usr/bin/ipcs64
- /usr/bin/iostat
- /usr/bin/logout
- /usr/bin/lscore
- /usr/bin/lssec
- /usr/bin/mesg
- /usr/bin/mkgroup
- /usr/bin/mkque
- /usr/bin/mkquedev
- /usr/bin/mkrole
- /usr/bin/mkuser
- /usr/bin/netpmon
- /usr/bin/newgrp
- /usr/bin/pagdel
- /usr/bin/paginit
- /usr/bin/paglist
- /usr/bin/passwd
- /usr/bin/pwck
- /usr/bin/pwdadm
- /usr/bin/pwdck
- /usr/bin/rm_mlcache_file
- /usr/bin/rdist
- /usr/bin/remsh
- /usr/bin/rlogin
- /usr/bin/rexec

- /usr/bin/rcp
- /usr/bin/rmque
- /usr/bin/rmquedev
- /usr/bin/rsh
- /usr/bin/ruptime
- /usr/bin/rwho
- /usr/bin/script
- /usr/bin/setgroups
- /usr/bin/setsenv
- /usr/bin/shell
- /usr/bin/su
- /usr/bin/sysck
- /usr/bin/tcbck
- /usr/bin/sysck_r
- /usr/bin/telnet
- /usr/bin/tftp
- /usr/bin/traceroute
- /usr/bin/tn
- /usr/bin/tn3270
- /usr/bin/usrck
- /usr/bin/utftp
- /usr/bin/vmstat
- /usr/bin/vmstat64
- /usr/bin/yppasswd
- /sbin/helpers/jfs2/backbyinode
- /sbin/helpers/jfs2/diskusg
- /sbin/helpers/jfs2/restbyinode

Felhasználók, csoportok és jelszavak

Kezelheti az AIX felhasználókat és csoportokat.

Saját könyvtár automatikus létrehozása bejelentkezéskor

Az AIX operációs rendszer automatikusan létrehozhat egy saját könyvtárat felhasználói bejelentkezés során.

Ez a szolgáltatás távolról megadott felhasználók (például az LDAP szerveren megadott felhasználók) esetén hasznos, akik esetlegesen nem rendelkeznek saját könyvtárral a helyi rendszeren. Az AIX operációs rendszer két módszert biztosít a saját könyvtár létrehozásához a felhasználó bejelentkezésekor: szabványos AIX mechanizmust és egy PAM mechanizmust. Ezek a mechanizmusok engedélyezhetők együtt.

AIX mechanizmus

Az AIX mechanizmus a következő parancsokon keresztüli bejelentkezést foglalja magában: **getty**, **login**, **rlogin**, **rsh**, **telnet** és **tsm**. Az AIX mechanizmus az STD_AUTH hitelesítést és a PAM_AUTH hotelesítést támogatja a pam_aix modul használatával. Engedélyezze az AIX mechanizmust az **/etc/security/login.cfg** fájlban az usw szakasz mkhomeatlogin attribútumának true értékre állításával (a fájllal kapcsolatos további információkat az /etc/security/login.cfg fájl tartalmaz). A **chsec** parancs segítségével engedélyezheti vagy letilthatja a saját könyvtát automatikus létrehozása bejelentkezéskor szolgáltatást. A szolgáltatás engedélyezéséhez például futtassa a következő parancsot:

```
# chsec -f /etc/security/login.cfg -s usw -a mkhomeatlogin=true
```

Ha engedélyezett, akkor a bejelentkezési folyamat a sikeres hitelesítés után ellenőrzi a felhasználó saját könyvtárát. Ha a felhasználó saját könyvtára nem létezik, akkor a rendszer létrehoz egyet.

Megjegyzés: Az **mkhomeatlogin** attribútum csak AIX Version 6.1 with the 6100-02 Technology Level vagy újabb rendszeren támogatott.

PAM mechanizmus

Az AIX egy pam_mkuserhome modult is biztosít a saját könyvtárak létrehozásához PAM mechanizmusok esetén. A pam_mkuserhome modul összefogható más munkamenet modulokkal bejelentkezési szolgáltatások esetén. A PAM modul engedélyezéséhez a szolgáltatáshoz egy bejegyzést hozzá kell adni ehhez a szolgáltatáshoz. A saját könyvtár létrehozás **telnet** parancson és PAM-on keresztüli engedélyezéséhez például adja hozzá a következő bejegyzést az **/etc/pam.cfg** fájlhoz:

```
telnet session optional pam_mkuserhome
```

Fiókazonosító

Minden felhasználói fiók rendelkezik egy numerikus azonosítóval, amely a fiókot azonosítja. Az AIX operációs rendszer a fiókazonosító szerint ad jogosultságot.

Fontos megérteni, hogy az ugyanazzal az azonosítóval rendelkező fiókok virtuálisan ugyanazt a fiókot jelentik. Felhasználók és csoportok létrehozásakor az AIX **mkuser** és **mkgroup** parancs mindig ellenőrzi a célnyilvántartásban, hogy a létrehozandó fiók azonosítója nem ütközik meglévő fiókokéval.

A rendszer úgy is beállítható, hogy a fióklétrehozás során az összes felhasználói (csoport) nyilvántartást ellenőrizze a **dist_uniqid** rendszerattribútum segítségével. A /etc/security/login.cfg fájlban az usw szakasz **dist_uniqid** attribútuma a **chsec** parancs segítségével kezelhető. Annak beállításához, hogy a rendszer mindig minden nyilvántartásban ellenőrizze az azonosítóütközést, futtassa az alábbi parancsot:

```
# chsec -f /etc/security/login.cfg -s usw -a dist_uniqid=always
```

A **dist_uniqid** attribútumnak három érvényes értéke van:

never Ez az érték csak a célnyilvántartásokban ellenőrzi az azonosítóütközést (ez az alapértelmezett).

always Ez az érték az összes többi nyilvántartásban is ellenőrzi az azonosítóütközést. Ha a cél- és más rendszerleíró nyilvántartás között ütközés történik, akkor az **mkuser** (**mkgroup**) parancs egy egyedi azonosítót vesz, amelyet egyik nyilvántartás sem használ. Ez csak akkor nem sikerül, ha az azonosítóérték parancssorból van megadva (például mkuser id=234 foo, és a 234 azonosítót az egyik nyilvántartásban lévő felhasználó már használja).

uniqbyname

Ez az érték az összes többi nyilvántartásban is ellenőrzi az azonosítóütközést. A nyilvántartások közötti ütközés csak akkor megengedett, ha a létrehozandó fiók ugyanazzal a névvel rendelkezik, mint a meglévő fiók mkuser id=123 foo típusú parancs esetén. Ha az azonosító nem parancssorban van megadva, akkor elképzelhető, hogy az új fiók nem ugyanazzal az azonosítóval rendelkezik, mint a másik nyilvántartásban lévő, azonos nevű meglévő fiók. A 234 azonosítóval rendelkező *acct1* például helyi fiók. *acct1* nevű LDAP fiók létrehozásakor előfordulhat, hogy az mkuser -R LDAP acct1 parancs a 235 egyedi azonosítót adja meg az LDAP fiókhoz. Ennek eredményeképp lesz egy 234 azonosítóval rendelkező *acct1* helyileg, meg egy 235 azonosítóval rendelkező *acct1* az LDAP címtáron.

Megjegyzés: A célnyilvántartásban az ütközésfelismerés mindig végrehajtásra kerül, a **dist_uniqid** attribútumtól függetlenül.

A **uniqbyname** érték két nyilvántartás esetén jól működik. Több nyilvántartás esetén, és ha a két nyilvántartás között már azonosítóütközés áll fenn, akkor az **mkuser** (**mkgroup**) viselkedése meghatározatlan lesz, ha egy harmadik nyilvántartásban az ütköző azonosítóértékekkel kerül létrehozásra új fiók. Az új fiók létrehozása a nyilvántartások ellenőrzésének sorrendjétől függően sikeres vagy sikertelen lehet.

Például: Tegyük fel, hogy a rendszer három nyilvántartással rendelkezik: helyi, LDAP és DCE. Egy *acct1* fiók létezik az LDAP és egy *acct2* a DCE fiókban, és mindkettő azonosítója 234. Ha a rendszeradminisztrátor az mkuser -R files

id=234 acct1 (mkgroup -R files id=234 acct1) parancs futtatásával létrehoz egy helyi fiókot **uniqbyname** értékkel, akkor az **mkuser** (**mkgroup**) parancs először az LDAP nyilvántartást ellenőrzi és azt találja, hogy a 234 azonosítót az *acct1* LDAP fiók használja. Mivel a létrehozandó fiók ugyanazzal a fióknévvel rendelkezik, az **mkuser** (**mkgroup**) parancs sikeresen létrehozza a helyi *acct1* fiókot a 234 azonosítóval. Ha először a DCE nyilvántartás kerül ellenőrzésre, akkor az **mkuser** (**mkgroup**) parancs azt találja, hogy a 234 azonosítót az *acct2* DCE fiók használja és az *acct1* létrehozása meghiúsul. Az azonosítóütközés ellenőrzése a helyi és távoli nyilvántartások közötti, vagy a távoli nyilvántartások közötti azonosítóegyediséget kényszerít ki. A távoli nyilvántartáson újonnan létrehozott és más rendszer meglévő helyi felhasználó azonosítójának egyediségére nincs garancia, amennyiben ugyanazt a nevet használják. Az **mkuser** (**mkgroup**) parancs kihagyja a távoli nyilvántartást, ha a parancs futtatásakor nem érhető el.

Root fiók

A root fiók lényegében korlátlan hozzáféréssel bír a rendszer programjaihoz, fájljaihoz és erőforrásaihoz.

A root fiók egy speciális felhasználó az /etc/passwd fájlban. Felhasználói azonosítója (UID) 0, a felhasználó neve pedig általában *root*. Nem a felhasználói név teszi olyan különlegessé a root fiókot, hanem a 0-ás UID. Ez azt is jelenti, hogy tetszőleges 0-ás UID értékkel rendelkező felhasználó szintén birtokolja a root felhasználó privilégiumait. Emellett a root fiók mindig a helyi biztonsági fájlok alapján kerül hitelesítésre.

A root fióknak mindig rendelkeznie kell jelszóval, és ezt a jelszót sohasem szabad megosztani. A root fióknak közvetlenül a rendszer telepítése után be kell állítani egy jelszót. A root jelszót csak a rendszer adminisztrátorának kell tudnia. A rendszeradminisztrátoroknak is csak akkor szabad root felhasználóként tevékenykedniük, ha a végzett adminisztrátori funkció root privilégiumokat igényel. Minden más művelethez vissza kell térniük szokásos felhasználói fiókjukhoz.

FIGYELEM: A root felhasználói fiók rutinszerű használata a rendszer sérüléséhez is vezethet, mivel a root fiók a rendszer számos biztonsági funkcióját hatálytalanítja.

Közvetlen root bejelentkezés letiltása:

A lehetséges betörők egyik kedvelt támadási formája a root felhasználó jelszavának megszerzése.

Ez ilyen típusú támadások kivédéséhez érdemes letiltani a root azonosító közvetlen elérését, ily módon ugyanis a rendszeradminisztrátor csak az **su -** paranccsal tud root felhasználóként bejelentkezni. A root felhasználó közvetlen hozzáférésének letiltása az említett támagások lehetőségének kivédése mellett azzal az előnnyel is jár, hogy pontosan megfigyelheti, mely felhasználók nyertek root felhasználói privilégiumokat és mikor. Ezt a /var/adm/sulog fájl megtekintésével érheti el. Egy másik lehetőség a rendszer megfigyelésének ellenőrzése, amely jelenti az ilyen típusú tevékenységeket.

A root felhasználó távoli bejelentkezésének letiltásához módosítsa az /etc/security/user fájlt. A root bejegyzés rlogin attribútumában állítsa be a false értéket.

Mielőtt letiltaná a root távoli bejelentkezését, gondosan vizsgálja meg, hogy milyen helyzetek akadályozhatják meg a rendszeradminisztrátort a saját (nem root) felhasználói azonosítójával való belépésben. Ha például egy felhasználó saját fájlrendszere megtelt, akkor a felhasználó nem tud bejelentkezni. Ha a távoli root bejelentkezés le van tiltva, és a felhasználó, aki a **su -** parancs segítségével át tud váltani root felhasználó, teljes home fájlrendszerrel rendelkezik, akkor a root többé nem tudja irányítani a rendszert. Ez a probléma úgy kerülhető el, hogy a rendszeradminisztrátorok saját könyvtárát tartalmazó fájlrendszert nagyobbra állítjuk az átlagos felhasználók fájlrendszerénél.

Felhasználói fiókok

A felhasználói fiókokhoz számos biztonsági adminisztrációs feladat tartozik.

Ajánlott felhasználói attribútumok:

A felhasználói adminisztráció a felhasználók és csoportok létrehozásából és attribútumaik meghatározásából áll.

A felhasználók egyik leglényegesebb jellemzője a hitelesítésük módja. A felhasználók a rendszer elsődleges ügynökei. Attribútumaik hatással vannak a hozzáférési jogaikra, környezetükre, hitelesítésük módjára, illetve a fiókjuk elérésének időpontjára, helyére és módjára.

A csoport a védett erőforrásokhoz azonos hozzáférési engedélyekkel rendelkező felhasználók gyűjteménye. A csoportok rendelkeznek egy azonosítóval, és tagokból illetve adminisztrátorokból állnak. A csoport első adminisztrátora általában a csoport létrehozója.

Minden egyes felhasználói fiókhoz számos attribútum, például jelszó és bejelentkezési tulajdonságok állíthatók be. A beállítható attribútumokról további információkat a “Lemezkvótarendszer áttekintése” oldalszám: 74 szakaszban talál. A következő tulajdonságok beállítása javasolt:

- Minden felhasználónak egyedi felhasználói azonosítóval kell rendelkeznie. A biztonsági és elszámoltathatósági eszközök csak akkor működnek, ha minden felhasználónak egyedi azonosítója van.
- A felhasználói neveknek jelentéssel kell bírniuk a rendszer többi felhasználója számára. A legjobb a felhasználó tényleges neve, mivel a legtöbb elektronikus levelezési rendszer a bejövő postát a felhasználó azonosítója szerint címzi.
- Felhasználók hozzáadása, módosítása és törlése a SMIT felület használatával. Bár mindegyik feladat elvégezhető parancssorból is, a SMIT felület segítséget nyújt a kisebb hibák előfordulásának csökkentésében.
- Ne adjon a felhasználóknak kezdeti jelszót, amíg a felhasználó nem áll készen a bejelentkezésre. Ha egy felhasználó jelszava az /etc/passwd fájlban egy csillag (*), akkor a fiókinformációk aktívak, ugyanakkor senki nem tud bejelentkezni rá.
- Ne módosítsa a rendszer által meghatározott felhasználói azonosítókat, mivel ezek a rendszer megfelelő működéséhez szükségesek. A rendszer által meghatározott felhasználói azonosítók az /etc/passwd fájlban találhatók.
- Általában ne állítsa egyik felhasználói azonosító *admin* paraméterét se true értékre. Az /etc/security/user fájlban admin=true értékkel rendelkező felhasználók attribútumait csak a root felhasználó módosíthatja.

Az operációs rendszer a /etc/passwd és /etc/system/group fájlban található szabványos felhasználói attribútumokat támogatja, úgymint:

Hitelesítési információk
Megadja a jelszót

Azonosítók
Megadja a felhasználó azonosítóját, azonosítócsoportját és a kiegészítő csoportazonosítót.

Környezet
Megadja a saját vagy héj környezetet.

Felhasználó- és csoportnévhossz-korlát:

Beállíthatja és lekérheti a felhasználói és csoportnév hosszának korlátját.

A felhasználó- és csoportnévhossz-korlát paraméter alapértelmezett értéke 9 karakter. AIX 5.3 és újabb változat esetén növelheti a felhasználó- és csoportnévhossz-korlátot 9 karakterről 256 karakterre. Mivel a felhasználó- és csoportnévhossz-korlát paraméter tartalmazza a befejező NULL karaktert, a tényleges érvényes névhossz 8 - 255 karakter.

A felhasználó- és csoportnévhossz-korlát a **v_max_logname** rendszerkonfigurációs paraméterrel van megadva a sys0 eszközhöz. Módosíthatja vagy lekérheti a **v_max_logname** paraméterértéket a kernelről vagy az ODM adatbázisból. A kernelben lévő paraméterérték a rendszer által futás közben használt érték. Az ODM adatbázisban lévő paraméterérték a rendszer által a következő újraindítás után használt érték.

Megjegyzés: Váratlan viselkedés léphet fel, ha növelés után csökkenti a felhasználó- és csoportnévhossz korlátját. A rendszeren létezhetnek még a nagyobb korláttal létrehozott felhasználó- és csoportnevek.

Felhasználó- és csoportnévhossz-korlát lekérése az ODM adatbázisból:

Parancsokat és szubrutinokat használhat a **v_max_logname** paraméter lekéréséhez.

Az **lsattr** parancs segítségével lekérheti az ODM adatbázis **v_max_logname** paraméterét. Az **lsattr** parancs a **v_max_logname** paramétert max_logname attribútumként jeleníti meg.

További információkért tekintse meg az **lsattr** parancsot a következő témakörben: *Commands Reference, Volume 3*.

A következő példa bemutatja, hogy az **lsattr** parancs segítségével hogyan kérhető le a max_logname attribútum:

```
$ lsattr -El sys0
SW_dist_intr    false           Enable SW distribution of interrupts             True
autorestart     true            Automatically REBOOT system after a crash        True
boottype        disk            N/A                                              False
capacity_inc    1.00            Processor capacity increment                     False
capped          true            Partition is capped                              False
conslogin       enable          System Console Login                             False
cpuguard        enable          CPU Guard                                        True
dedicated       true            Partition is dedicated                           False
ent_capacity    4.00            Entitled processor capacity                      False
frequency       93750000        System Bus Frequency                             False
fullcore        false           Enable full CORE dump                            True
fwversion       IBM,SPH01316    Firmware version and revision levels             False
iostat          false           Continuously maintain DISK I/O history           True
keylock         normal          State of system keylock at boot time             False
max_capacity    4.00            Maximum potential processor capacity             False
max_logname     20              Maximum login name length at boot time           True
maxbuf          20              Maximum number of pages in block I/O BUFFER CACHE True
maxmbuf         0               Maximum Kbytes of real memory allowed for MBUFS  True
maxpout         0               HIGH water mark for pending write I/Os per file  True
maxuproc        128             Maximum number of PROCESSES allowed per user     True
min_capacity    1.00            Minimum potential processor capacity             False
minpout         0               LOW water mark for pending write I/Os per file   True
modelname       IBM,7044-270    Machine name                                     False
ncargs          6               ARG/ENV list size in 4K byte blocks              True
pre430core      false           Use pre-430 style CORE dump                      True
pre520tune      disable         Pre-520 tuning compatibility mode                True
realmem         3145728         Amount of usable physical memory in Kbytes       False
rtasversion     1               Open Firmware RTAS version                       False
sec_flags       0               Security Flags                                   True
sed_config      select          Stack Execution Disable (SED) Mode               True
systemid        IBM,0110B5F5F   Hardware system identifier                       False
variable_weight 0               Variable processor capacity weight               False
$
```

Felhasználó- és csoportnévhossz-korlát lekérése a kernelből:

Parancsokat és szubrutinokat használhat a **v_max_logname** paraméter kernelből való lekéréséhez.

A getconf parancs használata

A **getconf LOGIN_NAME_MAX** paraméterrel való kiadásával lekérdezheti a kernel felhasználó- és csoportnévhossz-korlátját. A **getconf** parancs kimenete tartalmazza a befejező NULL karaktert.

A következő példa bemutatja, hogy a **getconf** parancs segítségével hogyan kérhető le az aktuális felhasználó- és csoportnévkorlát a kernelből:

```
$ getconf LOGIN_NAME_MAX
20
$
```

A sysconf szubrutin használata

A sysconf szubrutin **_SC_LOGIN_NAME_MAX** paraméterrel való kiadásával lekérheti a kernel felhasználó- és csoportnévhossz-korlátját.

A következő példa bemutatja, hogy a sysconf szubrutin segítségével hogyan kérhető le a felhasználó- és csoportnévhossz-korlát a kernelből:

```
#include <unistd.h>
main()
{
       long len;

       len = sysconf(_SC_LOGIN_NAME_MAX);

       printf("The name length limit is %d\n", len);
}
```

A sys_parm szubrutin használata

A sys_parm szubrutin **SYSP_V_MAX_LOGNAME** paraméterrel való használatával lekérheti a kernel aktuális felhasználó- és csoportnévkorlátját.

A következő példa bemutatja, hogy a sys_parm szubrutin segítségével hogyan kérhető le a felhasználónévhossz-korlát a kernelből:

```
#include <sys/types.h>
#include <sys/var.h>
#include <errno.h>
main()
{
   int rc;
   struct vario myvar;

   rc = sys_parm (SYSP_GET, SYSP_V_MAX_LOGNAME, &myvar);

   if (!rc)
      printf("Max_login_name = %d\n", myvar.v.v_max_logname.value);
   else
      printf("sys_parm() failed rc = %d, errno = %d\n", rc, errno);
}
```

Felhasználói csoport- és névhosszkorlát módosítása az ODM adatbázisban:

A felhasználói és csoportnév hosszának korlátját a kernelben csak a rendszerbetöltési fázis során állíthatja be. Az ODM adatbázisban lévő értéket a **chdev** parancs segítségével módosíthatja. A módosítás a következő rendszer-újraindítás után lép életbe.

A következő példa bemutatja, hogy a **chdev** parancs segítségével hogyan módosítható a **v_max_logname** paraméter az ODM adatbázisban:

```
$ chdev -l sys0 -a max_logname=30
sys0 changed
$
```

Felhasználói fiók felügyelet:

A felhasználói fiókok attribútumokkal rendelkeznek, amelyek megváltoztathatók.

Minden felhasználói fiók rendelkezik egy sor társított attribútummal. Ezek az attribútumok az alapértelmezett értékek alapján jönnek létre, amikor a felhasználó az **mkuser** parancs segítségével létrehozásra kerül. Az attribútumok módosítására a **chuser** parancs használható. Az alábbiakban a bejelentkezést vezérlő, de nem a jelszó minőségével kapcsolatos attribútumok felsorolását találja:

account_locked
Ha egy fiókot kifejezetten zárolni kell, akkor az attribútum beállítható True értékre. Az alapértelmezés a False.

admin Ha értéke True, akkor a felhasználó nem módosíthatja jelszavát. Csak az adminisztrátor módosíthatja.

admgroups
Felsorolja azokat a csoportokat, amelyekben a felhasználónak adminisztrátori jogosultságai vannak. Ezekben a csoportokban a felhasználó jogosult tagok hozzáadására és eltávolítására.

auth1 A felhasználói hozzáférés biztosítására használt hitelesítési módszer. Értéke általában SYSTEM, amely újabb módszereket is használhat.

Megjegyzés: Az **auth1** attribútum elavult és nem használható.

auth2 A felhasználónak az **auth1** paraméterben megadott hitelesítése után lefutó módszer. Ez már nem akadályozhatja meg a rendszer elérését. Értéke általában NONE.

Megjegyzés: Az **auth2** attribútum elavult és nem használható.

daemon
Ez a logikai paraméter határozza meg, hogy a felhasználó jogosult-e démonok vagy alrendszerek indítására a **startsrc** paranccsal. A cron és at szolgáltatások használatát is korlátozza.

login Megadja, hogy a felhasználó jogosult-e a bejelentkezésre. A sikeres bejelentkezés az **unsuccessful_login_count** attribútum értékét visszaállítja 0-ra (a **loginsuccess** szubrutinból).

logintimes
Korlátozza a felhasználó bejelentkezési időszakát. Ezzel korlátozható például a felhasználó bejelentkezése a szokásos munkaidőre.

registry
Megadja a felhasználói nyilvántartást. Ezzel írható elő a rendszernek más nyilvántartás, például NIS, LDAP vagy Kerberos használata a felhasználói információkhoz.

rlogin Megadja, hogy a megadott felhasználó bejelentkezhet-e az **rlogin** vagy a **telnet** paranccsal. Az rlogin attribútum csak távoli bejelentkezést vezérel. Az egyedi távoli parancsok futtatási képességének szabályozásáról információkért lásd: rcmds.

su Megadja, hogy más felhasználók átválthatnak-e erre az azonosítóra a **su** paranccsal.

sugroups
Megadja, hogy mely csoportok tagjai válthatnak át erre a felhasználói azonosítóra.

ttys Ezzel korlátozhatók bizonyos fiókok fizikailag biztonságos területekre.

expires Segítséget nyújt a tanulói vagy vendég fiókok kezelésére, emellett ideiglenes letiltásra is használható.

loginretries
Megadja, hogy hány egymást követő sikertelen bejelentkezési kísérlet zárolja a felhasználói azonosítót a rendszerben. A sikertelen kísérleteket az /etc/security/lastlog tárolja.

umask Megadja a felhasználó kezdeti **umask** értékét.

rcmds Megadja, hogy a megadott felhasználó futtathat-e egyedi parancsokat az **rsh** paranccsal vagy az **rexec** paranccsal. Az allow érték jelzi azt, hogy futtathat parancsokat távoli módon az **rsh** és **rexec** parancsokkal. A deny érték azt jelzi, hogy nem futtathat parancsokat távoli módon. A hostlogincontrol érték azt jelzi, hogy a távoli parancsok futtatását a**hostallowedlogin** és a **hostsdeniedlogin** attribútum szabályozza. A távoli bejelentkezés vezérlésével kapcsolatos információkért lásd: rlogin attribútum.

hostallowedlogin
Megadja, hogy a felhasználó mely hosztokra jelentkezhet be. Ez az attribútum hálózati környezetben használható, ahol a felhasználói attribútumok több hoszt között oszlanak meg.

hostsdeniedlogin
Megadja, hogy a felhasználó mely hosztokra nem jelentkezhet be. Ez az attribútum hálózati környezetben használható, ahol a felhasználói attribútumok több hoszt között oszlanak meg.

maxulogs
Megadja a felhasználó maximális bejelentkezéseinek számát. Ha a felhasználó bejelentkezéseinek száma elérte ezt az értéket, akkor a további bejelentkezések nem engedélyezettek.

A felhasználói attribútumok teljes halmaza az /etc/security/user, /etc/security/limits, /etc/security/audit/config és /etc/security/lastlog fájlban van megadva. Az **mkuser** parancs alapértelmezéseit az /usr/lib/security/mkuser.default fájl adja meg. Az mkuser.default fájlban csak az /etc/security/user és /etc/security/limits fájlok általános alapértelmezett szakaszait felülbíráló beállításokat, illetve a megfigyelési osztályokat szabad megadni. Ezen attribútumok közül több is hatással van a felhasználói bejelentkezés módjára, emellett beállíthatók a felhasználói fiók automatikus zárolására (vagyis a bejelentkezés megakadályozására) a megadott feltételek teljesülése esetén.

Ha egy felhasználói fiókot a rendszer adott számú sikertelen bejelentkezési kísérlet miatt zárolt, akkor a felhasználó nem fog tudni bejelentkezni mindaddig, amíg a rendszeradminisztrátor át nem állítja a felhasználó **unsuccessful_login_count** attribútumát az /etc/security/lastlog fájlban egy olyan értékre, amely a bejelentkezési kísérletek megengedett számánál kisebb. Erre a **chsec** parancs használható az alábbiak szerint:

```
chsec -f /etc/security/lastlog -s felhasználónév -a
unsuccessful_login_count=0
```

Az alapértelmezett értékek módosításához a **chsec** parancs segítségével módosítani kell a megfelelő biztonsági fájl, például a /etc/security/user vagy /etc/security/limits fájl alapértelmezett szakaszát. Számos alapértelmezés szabványos viselkedésként került meghatározásra. Ha olyan attribútumokat kíván explicit módon megadni, amelyek új felhasználó létrehozásakor mindig beállításra kerülnek, akkor módosítsa a /usr/lib/security/mkuser.default fájl *user* bejegyzését.

A kiterjesztett jelszó attribútumokról további információkat a “Jelszavak” oldalszám: 62 szakaszban talál.

Felhasználói attribútumok által befolyásolt bejelentkezéssel kapcsolatos parancsok

Az alábbi táblázat a bejelentkezést és az érintett parancsokat vezérlő attribútumokat mutatja be.

Felhasználói attribútum	Parancsok
account_locked	**rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet**, **ftp**, **login**
login	Csak a konzolról történő bejelentkezésre van hatással. A **login** attribútum értéke nincs hatással a távoli bejelentkezési parancsokra, a távoli parancsértelmező parancsokra és a távoli másolás parancsokra (**rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet** és **ftp**).
logintimes	**rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet**, **ftp**, **login**
rlogin	Csak a távoli bejelentkezési parancsokra, bizonyos távoli parancsértelmező parancsokra és bizonyos távoli másolás parancsokra van hatással (**ssh**, **scp**, **rlogin** és **telnet**).
loginretries	**rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet**, **ftp**, **login**
/etc/nologin	**rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet**, **ftp**, **login**
rcmds=deny	**rexec**, **rsh**, **rcp**, **ssh**, **scp**
rcmds=hostlogincontrol és hostsdeniedlogin=<target_hosts>	**rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet**, **ftp**, **login**
ttys = !REXEC, !RSH	**rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet**, **ftp**, **login**
ttys = !REXEC, !RSH, /dev/pts	**rexec**, **rsh**
ttys = !REXEC, !RSH, ALL	**rexec**, **rsh**
expires	**rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet**, **ftp**, **login**

Megjegyzés: Az **rsh** csak a távoli parancsok végrehajtását tiltja le. A távoli bejelentkezések engedélyezettek.

Kapcsolódó tájékoztatás:
loginsuccess szubrutin
rexec parancs
rsh parancs
startsrc parancs
su parancs

Bejelentkezési felhasználói azonosítók:

Az operációs rendszer a bejelentkezési felhasználói azonosító alapján azonosítja a felhasználót.

A bejelentkezési felhasználói azonosító teszi lehetővé a rendszernek a felhasználói tevékenységek nyomon követését. Miután a felhasználó bejelentkezett a rendszerbe, de még mielőtt lefutna a kezdeti felhasználói program, a rendszer beállítja a folyamat bejelentkezési azonosítóját a felhasználói adatbázisban talált felhasználói azonosítóra. A bejelentkezési munkamenet minden további folyamata ezzel az azonosítóval kerül megjelölésre. Ezek a címkék biztosítják a bejelentkezési felhasználói azonosítóval végzett tevékenységek követhetőségét. A felhasználó a munkamenet során visszaállíthatja a hatályos felhasználói azonosítót, a valódi felhasználói azonosítót, a hatályos csoportazonosítót és a kiegészítő csoportazonosítót, de nem módosíthatja a bejelentkezési felhasználói azonosítót.

Felhasználói biztonság megerősítése hozzáférés felügyeleti listákkal:

A rendszerbiztonság megfelelő szintjének eléréséhez ki kell alakítani egy összefüggő biztonsági házirendet a felhasználói fiókok kezelésére. A legelterjedtebb biztonsági mechanizmus a hozzáférés felügyeleti lista (ACL).

A hozzáférés felügyeleti listákról és a biztonsági házirendek kialakításáról a kézikönyv “Hozzáférés felügyeleti listák” oldalszám: 118 szakaszában olvashat.

PATH környezeti változó:

A **PATH** környezeti változó egy fontos biztonsági elem. A parancsok keresésekor használt könyvtárakat határozza meg.

Az alapértelmezett rendszerszintű **PATH** érték az /etc/profile fájlban van megadva, emellett minden felhasználó rendelkezik egy saját **PATH** értékkel a saját $HOME/.profile fájlban. A .profile fájlban lévő **PATH** érték felülírja a rendszerszintű **PATH** értéket vagy további könyvtárakat ad hozzá.

A **PATH** környezeti változó jogosulatlan módosításai lehetővé tehetik a rendszer felhasználóinak más felhasználók "meghamisítását", és ez alól a root felhasználók sem kivételek. A *hamisító* (más néven *trójai*) programok lecserélik a rendszer parancsait, majd elfogják az eredeti parancsnak szánt információkat, például a felhasználói jelszavakat.

Tegyük fel például, hogy egy felhasználó úgy módosítja a **PATH** értékét, hogy a rendszer a parancsok futtatásakor először a /tmp könyvtárban keres. Ezután a felhasználó elhelyez a /tmp könyvtárban egy **su** nevű programot, amely az eredeti **su** paranccsal megegyező módon bekéri a root jelszót. Ezután a **/tmp/su** program elküldi a root jelszót a felhasználónak, és kilépés előtt meghívja a valódi **su** parancsot. A felvázolt példában az **su** parancsot használó valamennyi root felhasználó felfedte a jelszavát, méghozzá anélkül, hogy tudna erről.

A rendszeradminisztrátorok és felhasználók **PATH** környezeti változójával kapcsolatos problémák néhány egyszerű lépéssel elkerülhetők:

- Ha kétségei vannak, akkor használjon teljes útvonalakat. Teljes elérési út megadása esetén a **PATH** környezeti változó figyelmen kívül marad.
- Soha ne helyezze az aktuális könyvtárat (**.** (pont)) a root felhasználó **PATH** értékébe. Soha ne engedje az aktuális könyvtár megadását az /etc/profile fájlban.
- A root felhasználónak saját **PATH** meghatározással kell rendelkeznie a saját .profile fájljában. Az /etc/profile az összes felhasználó minimális igényét adja meg, míg a root felhasználónak ennél több könyvtárra lehet szüksége.

- Figyelmeztesse a felhasználókat, hogy ne módosítsák a .profile fájljukat a rendszeradminisztrátorral végzett konzultáció nélkül. Ellenkező esetben egy gyanútlan felhasználó nem kívánatos hozzáféréshez vezető módosításokat is végezhet. A felhasználók .profile fájlján 740 engedélyeket kell beállítani.
- A rendszeradminisztrátorok nem használhatják felhasználói szekcióból a **su** parancsot root jogosultságok szerzésére, mivel a felhasználónak a .profile fájlban beállított **PATH** értéke van hatályban. A felhasználók beállíthatják saját .profile fájljukat. A rendszeradminisztrátoroknak root felhasználóként kell bejelentkezniük a felhasználó gépére, vagy inkább a saját felhasználójukkal, majd ezután futtathatják az alábbi parancsot:

  ```
  /usr/bin/su - root
  ```

  Ez biztosítja, hogy a szekció a root környezetet fogja használni. Ha egy rendszeradminisztrátor másik felhasználói szekcióban tevékenykedik root felhasználóként, akkor a rendszeradminisztrátornak a szekcióban teljes elérési utakat kell használnia.
- Az /etc/profile fájl bemeneti mező elválasztó (**IFS**) környezeti változóját védeni kell a módosítással szemben. A .profile fájl **IFS** környezeti változója felhasználható a **PATH** érték megváltoztatására.

secldapclntd démon használata:

A **secldapclntd** démon dinamikusan kezeli az LDAP szerver kapcsolatait.

Induláskor a **secldapclntd** démon csatlakozik az **/etc/security/ldap/ldap.cfg** fájlban megadott szerverekhez (LDAP szerverenként egy kapcsolat). Ha később a **secldapclntd** démon meghatározza, hogy az LDAP kapcsolat korlátozza az LDAP feldolgozási kéréseket, akkor a démon automatikusan létesít egy másik kapcsolatot az aktuális LDAP szerverhez. Ez a folyamat a kapcsolatok előre meghatározott maximális számának eléréséig folytatódik. A kapcsolatok maximális számának elérése után nem kerül hozzáadásra kapcsolat.

A **secldapclntd** démon rendszeres időközönként ellenőrzi az aktuális LDAP szerver összes kapcsolatát. Ha egy kapcsolat - az első kapcsolat kivételével - tétlen egy előre meghatározott időtartamig, akkor a démon lezárja a kapcsolatot.

Az **/etc/security/ldap/ldap.cfg** fájl connectionsperserver változója a kapcsolatok maximális számaként kerül felhasználásra. Azonban ha a connectionsperserver változó nagyobb a numberofthread változónál, akkor a **secldapclntd** démon beállítja a connectionsperserver értéket numberofthread értékre. A connectionsperserver érvényes értékei a 1 - 100 tartományba esnek. Az alapértelmezett érték a 10 (connectionsperserver: 10).

A connectionmissratio változó az **/etc/security/ldap/ldap.cfg** fájlban beállítja az új LDAP kapcsolatok kialakításának feltételeit. A connectionmissratio változó adja meg, hogy a csatlakozások hány százalékánál nem sikerült az első alkalommal LDAP-kapcsolatot létesíteni. Ha a hibás kísérletek száma nagyobb a connectionmissratio változónál, akkor a **secldapclntd** démon kibővíti az LDAP lekérdezéseket új LDAP kapcsolatok létesítésével (nem a connectionsperserver változóban megadott kapcsolatok számának meghaladása). A connectionmissratio változó érvényes értékei a 10 - 90 tartományba esnek. Az alapértelmezett érték az 50 (connectionmissratio: 50).

Az **/etc/security/ldap/ldap.cfg** fájl connectiontimeout változója megadja, hogy a kapcsolatok meddig maradhatnak tétlenek, mielőtt a **secldapclntd** démon bezárná azokat. A connectiontimeout változó érvényes értékei az 5 másodperc fölötti értékek (nincs maximális korlát). Az alapértelmezett érték a 300 másodperc (connectiontimeout: 300).

Anonim FTP biztonságos felhasználói fiók beállításával

Beállíthat anonim FTP hozzáférést biztonságos felhasználói fiókkal.

Ez a példahelyzet a parancssori felület és egy parancsfájl segítségével anonim FTP hozzáférést állít be biztonságos felhasználói fiókkal.

1. A következő parancs beírásával ellenőrizze, hogy a bos.net.tcp.client fájlkészlet telepítve van-e a rendszerre:

   ```
   lslpp -L | grep bos.net.tcp.client
   ```

   Ha nincs kimenet, akkor a fájlkészlet nincs telepítve. A telepítéshez olvassa el a *Installation and migration* dokumentumot.
2. Root jogosultsággal lépjen be a /usr/samples/tcpip könyvtárba. Például:

```
cd /usr/samples/tcpip
```

3. A fiók beállításához futtassa a következő parancsfájlt:
   ```
   ./anon.ftp
   ```
4. Amikor az Are you sure you want to modify /home/ftp? (Biztosan módosítja a /home/ftp könyvtárt?) kérdés megjelenik, írja be a yes (igen) választ. Az alábbihoz hasonló kimenet jelenik meg:
   ```
   Added user anonymous.
   Made /home/ftp/bin directory.
   Made /home/ftp/etc directory.
   Made /home/ftp/pub directory.
   Made /home/ftp/lib directory.
   Made /home/ftp/dev/null entry.
   Made /home/ftp/usr/lpp/msg/en_US directory.
   ```
5. Lépjen be a /home/ftp könyvtárba. Például:
   ```
   cd /home/ftp
   ```
6. Hozzon létre home alkönyvtárat a következő beírásával:
   ```
   mkdir home
   ```
7. Módosítsa a /home/ftp/home könyvtár engedélyeit drwxr-xr-x értékre a következő beírásával:
   ```
   chmod 755 home
   ```
8. Lépjen be a /home/ftp/etc könyvtárba a következő beírásával:
   ```
   cd /home/ftp/etc
   ```
9. Hozza létre az objrepos alkönyvtárt a következő beírásával:
   ```
   mkdir objrepos
   ```
10. Módosítsa a /home/ftp/etc/objrepos könyvtár engedélyeit drwxrwxr-x értékre a következő beírásával:
    ```
    chmod 775 objrepos
    ```
11. Módosítsa a /home/ftp/etc/objrepos könyvtár tulajdonosát és csoportját a root felhasználóra és a system csoportra a következő beírásával:
    ```
    chown root:system objrepos
    ```
12. Hozzon létre egy security alkönyvtárt a következő beírásával:
    ```
    mkdir security
    ```
13. Módosítsa a /home/ftp/etc/security könyvtár engedélyeit drwxr-x--- értékre a következő beírásával:
    ```
    chmod 750 security
    ```
14. Módosítsa a /home/ftp/etc/security könyvtár tulajdonosát és csoportját a root felhasználóra és a security csoportra a következő beírásával:
    ```
    chown root:security security
    ```
15. Lépjen be a /home/ftp/etc/security könyvtárba a következő beírásával:
    ```
    cd security
    ```
16. Vegyen fel egy felhasználót a következő SMIT gyorselérés beírásával:
    ```
    smit mkuser
    ```

    Ebben a példahelyzetben egy test nevű felhasználót veszünk fel.
17. A SMIT mezőkben adja meg a következő értékeket:
    ```
    User NAME                                       [test]
    ADMINISTRATIVE USER?                             true
    Primary GROUP                                   [staff]
    Group SET                                       [staff]
    Another user can SU TO USER?                     true
    HOME directory                                  [/home/test]
    ```

    Miután megadta a változtatásokat, az Enter megnyomásával hozza létre a felhasználót. A SMIT folyamat befejeződése után lépjen ki a SMIT alkalmazásból.
18. Hozzon létre jelszót a felhasználó számára a következő paranccsal:

```
passwd test
```

Amikor a program felszólítja, adja meg a kívánt jelszót. Megerősítés céljából meg kell adnia másodszor is az új jelszót.

19. Lépjen be a /home/ftp/etc könyvtárba a következő beírásával:
    ```
    cd /home/ftp/etc
    ```
20. Másolja a /etc/passwd fájlt a /home/ftp/etc/passwd fájlba a következő paranccsal:
    ```
    cp /etc/passwd /home/ftp/etc/passwd
    ```
21. A kedvenc szövegszerkesztőjével szerkessze a /home/ftp/etc/passwd fájlt. Például:
    ```
    vi passwd
    ```
22. Távolítsa el az átmásolt tartalom összes sorát kivéve a root, ftp és test felhasználókét. A szerkesztés után a tartalomnak a következőhöz kell hasonlítania:
    ```
    root:!:0:0::/:/bin/ksh
    ftp:*:226:1::/home/ftp:/usr/bin/ksh
    test:!:228:1::/home/test:/usr/bin/ksh
    ```
23. Mentse el a módosításokat és lépjen ki a szerkesztőből.
24. Módosítsa a /home/ftp/etc/passwd fájl engedélyeit -rw-r--r-- értékre a következő beírásával:
    ```
    chmod 644 passwd
    ```
25. Módosítsa a /home/ftp/etc/passwd fájl tulajdonosát és csoportját a root felhasználóra és a security csoportra a következő beírásával:
    ```
    chown root:security passwd
    ```
26. Másolja a /etc/security/passwd fájl tartalmát a /home/ftp/etc/security/passwd fájlba a következő paranccsal:
    ```
    cp /etc/security/passwd /home/ftp/etc/security/passwd
    ```
27. A kedvenc szövegszerkesztőjével szerkessze a /home/ftp/etc/security/passwd fájlt. Például:
    ```
    vi ./security/passwd
    ```
28. Távolítson el minden szakaszt az átmásolt tartalomból a test felhasználó szakaszának kivételével.
29. Távolítsa el a flags = ADMCHG sort a test felhasználó szakaszából. A módosítás után a tartalomnak a következőhöz kell hasonlítania:
    ```
    test:
            password = 2HaAYgpDZX3Tw
            lastupdate = 990633278
    ```
30. Mentse el a módosításokat és lépjen ki a szerkesztőből.
31. Módosítsa a /home/ftp/etc/security/passwd fájl engedélyeit -rw------- értékre a következő beírásával:
    ```
    chmod 600 ./security/passwd
    ```
32. Módosítsa a /home/ftp/etc/security/passwd fájl tulajdonosát és csoportját a root felhasználóra és a security csoportra a következő beírásával:
    ```
    chown root:security ./security/passwd
    ```
33. A kedvenc szövegszerkesztőjével hozza létre és szerkessze a /home/ftp/etc/group fájlt. Például:
    ```
    vi group
    ```
34. Vegye fel a következő sorokat a fájlba:
    ```
    system:*:0:
    staff:*:1:test
    ```
35. Mentse el a módosításokat és lépjen ki a szerkesztőből.
36. Módosítsa a /home/ftp/etc/group fájl engedélyeit -rw-r--r— értékre a következő beírásával:
    ```
    chmod 644 group
    ```
37. Módosítsa a /home/ftp/etc/group fájl tulajdonosát és csoportját a root felhasználóra és a security csoportra a következő beírásával:
    ```
    chown root:security group
    ```
38. A kedvenc szövegszerkesztőjével hozza létre és szerkessze a /home/ftp/etc/security/group fájlt. Például:

```
vi ./security/group
```

39. Vegye fel a következő sorokat a fájlba:

```
system:
 admin = true
staff
 admin = false
```

40. Mentse el a módosításokat és lépjen ki a szerkesztőből. Ehhez tegye a következőket:
    a. Másolja a /etc/security/user fájlt a /home/ftp/etc/security könyvtárba a következő beírásával:

    ```
    cp /etc/security/user /home/ftp/etc/security
    cd /home/ftp/etc/
    ```

    b. A test felhasználó szakaszának kivételével távolítson el minden szakaszt az átmásolt tartalomból a szerkesztő segítségével a következő beírásával:

    ```
    vi ./security/user
    ```

    c. Mentse el a módosításokat és lépjen ki a szerkesztőből.
41. Módosítsa a /home/ftp/etc/security/group fájl engedélyeit -rw-r----- értékre a következő beírásával:

    ```
    chmod 640 ./security/group
    ```

42. Módosítsa a /home/ftp/etc/security/group fájl tulajdonosát és csoortját a root felhasználóra és a security csoportra a következő beírásával:

    ```
    chown root:security ./security/group
    ```

43. A következő parancsokkal másolja át a megfelelő tartalmat a /home/ftp/etc/objrepos könyvtárba:

    ```
    cp /etc/objrepos/CuAt ./objrepos
    cp /etc/objrepos/CuAt.vc ./objrepos
    cp /etc/objrepos/CuDep ./objrepos
    cp /etc/objrepos/CuDv ./objrepos
    cp /etc/objrepos/CuDvDr ./objrepos
    cp /etc/objrepos/CuVPD ./objrepos
    cp /etc/objrepos/Pd* ./objrepos
    ```

44. Lépjen be a /home/ftp/home könyvtárba a következő beírásával:

    ```
    cd ../home
    ```

45. Hozzon létre új saját könyvtárt a felhasználó számára a következő beírásával:

    ```
    mkdir test
    ```

    Ez lesz az új ftp felhasználó saját könyvtára.
46. Módosítsa a /home/ftp/home/test könyvtár tulajdonosát és csoportját a test felhasználóra és a staff csoportra a következők beírásával:

    ```
    chown test:staff test
    ```

47. Módosítsa a /home/ftp/home/test fájl engedélyeit -rwx------ értékre a következő beírásával:

    ```
    chmod 700 test
    ```

48. Tiltsa le a távoli bejelentkezést és a konzol bejelentkezést a test felhasználó számára az alábbi parancs beírásával:

    ```
    chuser login=false rlogin=false test
    ```

Ennél a pontnál be van állítva az ftp albejelentkezés a számítógépen. Ezt a következő eljárással próbálhatja ki:

1. Ftp segítségével kapcsolódjon a hosztgéphez, amelyen létrehozta a test felhasználót. Például:

   ```
   ftp MyHost
   ```

2. Jelentkezzen be anonymous felhasználóként. Amikor a rendszer kéri a jelszót, nyomja meg az Enter gombot.
3. Váltson az újonnan létrehozott test felhasználóra a következő paranccsal:

   ```
   user test
   ```

   Amikor a rendszer kéri a jelszót, használja az itt létrehozott jelszót: 18 oldalszám: 56.
4. A **pwd** paranccsal ellenőrizze, hogy létezik-e a felhasználó saját könyvtára. Például:

```
ftp> pwd
     /home/test
```

A kimenet a /home/test könyvtárat mutatja ftp alkönyvtárként. A hosztgépen a teljes útvonalnév valójában /home/ftp/home/test.

Notes:

- Csak ftp alfelhasználókkal rendelkező felhasználókra válthat át. Például a test egy ftp alfelhasználó.
- Amikor ftp anonymous felhasználókat hoz létre a anon.users.ftp parancsfájllal, a felhasználóhoz bármilyen nevet hozzárendelhet a *username* behelyettesítésével a parancsfájlban.
- Az anonymous felhasználók esetén, mivel a szerver a **chroot** parancsot a felhasználói fiók saját könyvtárában hajtja végre, minden konfigurációval kapcsolatos fájlnak, például a *fileftpaccess.ctl* fájlnak az adott felhasználó saját könyvtárában kell lennie, ami lehet például ~/etc/. Az /etc/ftpaccess.ctl fájlban a csak írási, írásvédett és olvasási/írási korlátozások útvonalának a chroot útvonalhoz relatívnak kell lennie.

További információk:

- Az *Security* című kiadvány "TCP/IP biztonság" szakasza.
- "**ftp parancs**" itt: *Commands Reference*

Rendszer-speciális felhasználói fiókok

Az AIX alapértelmezésben biztosít néhány speciális rendszer felhasználói fiókot, hogy ne a root és system felhasználói fiók tulajdonában legyen az operációs rendszer összes fájlja és fájlrendszere.

FIGYELEM: A rendszer speciális felhasználói fiókjainak eltávolításakor rendkívüli óvatossággal járjon el. Egy adott fiókot úgy tilthat le, hogy az /etc/security/passwd fájlban a megfelelő sor elejére beszúr egy csillagot (*). Vigyázzon azonban, nehogy letiltsa a root felhasználói fiókot. A rendszer speciális felhasználói fiókjainak eltávolításakor, vagy a root fiók letiltásakor az operációs rendszer nem fog működni.

Az operációs rendszer az alábbi előre meghatározott fiókokat tartalmazza:

adm Az adm felhasználói fiók a következő alapvető rendszerfunkciókat birtokolja:
- Az /usr/sbin/perf/diag_tool könyvtárban található diagnosztikai eszközöket.
- Az alábbi könyvtárakban tárolt elszámolási eszközöket:
  - /usr/sbin/acct
  - /usr/lib/acct
  - /var/adm
  - /var/adm/acct/fiscal
  - /var/adm/acct/nite
  - /var/adm/acct/sum

bin A bin felhasználói fiók birtokolja a legtöbb felhasználói parancs végrehajtható fájljait. A fiók elsődleges célja a fontosabb rendszerkönyvtárak és -fájlok tulajdonjogának megosztása, hogy ne minden fájlt a root és sys felhasználói fiókok birtokoljanak.

daemon
A daemon felhasználói fiók csak a rendszer szerver folyamatainak birtoklását és futtatását, illetve a társított fájlok birtoklását végzi. Ez a fiók garantálja, hogy a folyamatok futtatása a megfelelő fájlhozzáférési engedélyekkel történik.

nobody
A nobody felhasználói fiókot a Hálózati fájlrendszer (NFS) használja a távoli nyomtatás biztosításához. Ez a fiók azért van, hogy egy program ideiglenes root hozzáférést biztosíthasson a root felhasználóknak. A Biztonságos RPC vagy Biztonságos NFS engedélyezése előtt például ellenőrizze a mester NIM szerver /etc/public kulcsát annak meghatározásához, hogy mely felhasználóhoz nincs rendelve nyilvános kulcs és

titkos kulcs. Root felhasználóként létrehozhat egy bejegyzést az adatbázisban a hozzárendelésekkel nem rendelkező felhasználók számára a következő paranccsal:

```
newkey -u felhasználónév
```

Alternatív megoldásként létrehozhat egy bejegyzést az adatbázisban a nobody felhasználói fiók számára, így minden felhasználó futtathatja a **chkey** programot saját bejegyzéseinek létrehozásához anélkül, hogy ez root bejelentkezést igényelne.

root A root felhasználói fiók, 0-ás UID értékkel; ez a fiók használható a rendszerkarbantartási feladatok végrehajtásakor és a rendszerrel kapcsolatos problémák hibaelhárításakor.

sys A sys felhasználó birtokolja az Elosztott fájlszolgáltatás (DFS) gyorsítótár alapértelmezett felkapcsolási pontját, amelynek léteznie kell a DFS kliens telepítése vagy beállítása előtt. Az /usr/sys könyvtár tárolja emellett telepítőkészleteket.

system A system a rendszer által a rendszeradminisztrátorok számára létrehozott csoport. A csoport tagjai root jogosultság nélkül végrehajthatnak bizonyos karbantartási feladatokat.

Szükségtelen alapértelmezett felhasználói fiókok eltávolítása:

Az operációs rendszer telepítése során számos alapértelmezett felhasználói- és csoportazonosító létrejön. Az adott környezetben használt alkalmazásoktól illetve a rendszer hálózati helyétől függően ezen felhasználói- és csoportazonosítók egy része biztonsági kockázatot jelenthet.

Az alábbi táblázat sorolja fel azon általános alapértelmezett felhasználói azonosítókat, amelyeket esetleg el lehet távolítani:

3. táblázat: Eltávolítható általános alapértelmezett felhasználói azonosítók

Felhasználói azonosító	Leírás
uucp, nuucp	Az uucp protokoll által használt rejtett fájlok tulajdonosa. Az uucp felhasználói fiókot a UNIX-UNIX másoló program használja. Ez az AIX rendszerek legtöbbjén meglévő parancsokból, programokból és fájlokból álló csoport, amely lehetővé teszi a felhasználó számára, hogy dedikált vonalon vagy telefonvonalon keresztül kommunikáljon más AIX rendszerrel.
lpd	A nyomtatási alrendszer fájljainak tulajdonosa
guest	Lehetővé teszi fiókkal nem rendelkező felhasználók hozzáférését

Az alábbi táblázat sorolja fel azon általános csoportazonosítókat, amelyeket esetleg el lehet távolítani.

4. táblázat: Bizonyos helyzetekben szükségtelen általános csoportazonosítók

Csoportazonosító	Leírás
uucp	A uucp és nuucp felhasználók csoportja
printq	Az lpd felhasználó csoportja

A szükségtelen azonosítók meghatározásához elemezze a rendszert. Az említetteken felül esetleg további felhasználói- és csoportazonosítókat is el lehet távolítani. A rendszer éles környezetbe állítása előtt végezzen gondos kiértékelést a rendelkezésre álló azonosítókon.

Megjegyzés: A printq csoport eltávolítása helyett a printer fájlkészletektől való függőség miatt, tiltsa le az lp felhasználói azonosítót, a **piobe** parancsot és a qdaemon programot a /etc/inittab bejegyzésben a biztonsági kockázatok minimálisra csökkentése érdekében. Ez megakadályozza, hogy a felhasználó **print** parancsokat futtasson.

Biztonsági komponensekkel létrehozott fiókok:

Ha biztonsági komponensek például LDAP vagy OpenSSH van telepítve és beállítva, akkor a rendszer csoport azonosítókat hoz létre.

A rendszer az alábbi felhasználói és csoport azonosítókat hozza létre:

- **Internet protokoll (IP) biztonság:** Az IP biztonság az *ipsec* felhasználót és az *ipsec* csoportot adja hozzá a telepítéskor. Ezek az azonosítókat a kulcskezelő szolgáltatás használja. Ne feledje, hogy az /usr/lpp/group.id.keymgt helyen lévő csoport azonosítót nem lehet testreszabni a telepítés előtt.
- **Kerberos és Nyilvános kulcs infrastruktúra (PKI):** Ezek a komponensek nem hoznak létre új felhasználói vagy csoport fiókokat.
- **LDAP:** Ha az LDAP kliens és szerver telepítve van, akkor a rendszer létrehozza az *ldap* felhasználói azonosítót. Az *ldap* felhasználói azonosító nem rögzített. Az LDAP szerver telepítése automatikusan telepíti a DB2 adatbázist. A DB2 telepítés létrehozza a *dbsysadm* csoport azonosítót. A *dbsysadm* alapértelmezett csoport azonosítója a 400. Az LDAP szerver beállításakor az **mksecldap** parancs létrehozza az *ldapdb2* felhasználói fiókot.
- **OpenSSH:** Az OpenSSH telepítése hozzáadja az *sshd* és az *sshd* felhasználókat a rendszerhez. A megfelelő felhasználói- és csoport azonosítókat nem lehet módosítani. Az SSH privilégium elkülönítése használja az azonosítókat.

Tartomány nélküli csoportok

A tartomány nélküli csoportok szolgáltatás lehetővé teszi az egyik tartományban meghatározott felhasználók hozzárendelését másik tartományban meghatározott csoportokhoz. Ez a szolgáltatás csak LDAP protokollt és helyi tartományokat támogat.

Az LDAP szerveren felhasználókat és csoportokat létrehozhat az LDAP LDAP hitelesítés betöltési modul (LDAP modul) használatával. Felhasználókat és csoportokat a helyi rendszeren a Helyi hitelesítés betöltési modullal (helyi modul) is létrehozhat. Ha a **domainlessgroups** szolgáltatás nem engedélyezett, akkor az LDAP címtárban vagy a helyi rendszeren létrehozott felhasználók és felhasználói csoportok nem rendelhetők hozzá csoportokhoz a betöltési tartományon kívül, melyeken létre lettek hozva. Például az LDAP tartományban létrehozott felhasználó nem rendelhető hozzá a helyi tartományhoz hozzárendelt csoporthoz.

Ezt a korlátozást kikerülheti, és hozzárendelhet felhasználókat mind LDAP, mind helyi csoportokhoz a **domainlessgroups** rendszertulajdonság engedélyezésével. A **domainlessgroups** tulajdonság az /etc/secvars.cfg fájlban van meghatározva. Ez csak az LDAP és helyi modulok esetén támogatott. A tulajdonság lehetséges értékei:

false (alapértelmezett érték)
A csoport attribútum nincs összevonva az LDAP modulokból és helyi modulokból.

true A csoport attribútum össze van vonva az LDAP modulokból és helyi modulokból. Például az LDAP felhasználók hozzárendelhetők a helyi csoportokhoz.

A **domainlessgroups** tulajdonság értékének megjelenítéséhez futtassa a következő parancsot:

```
lssec -f /etc/secvars.cfg -s groups -a domainlessgroups
```

A **domainlessgroups** tulajdonság true értékre állításához futtassa a következő parancsot:

```
chsec -f /etc/secvars.cfg -s groups -a domainlessgroups=true
```

Az alábbi táblázat bemutatja, miben különböznek a felhasználói és csoport parancsok eredményei, a **domainlessgroups** tulajdonság beállításától függően.

5. táblázat: A kiválasztott parancsok **domainlessgroups** *tulajdonság által befolyásolt eredményei*

Parancs	Eredmény a domainlessgroups tulajdonság true értéke esetén
chgroup -R **ldap**|**files**	Frissíti a csoportot a megadott tartományban. Hozzáadhatja a felhasználót egy LDAP csoporthoz vagy helyi csoporthoz.
chuser -R **ldap**|**files**	Módosítja egy felhasználó beállításait a megadott tartományban. Ha a másik tartományban meghatározott csoportok meg vannak adva, akkor azok a csoportok szintén frissítve lesznek a felhasználói információkkal.
login *felhasználónév* vagy su	Lekéri a felhasználói attribútumokat a felhasználói nyilvántartásból, a csoportazonosító attribútum kivételével. A csoportazonosító felhasználói attribútumai összevonásra kerülnek mind az LDAP, mind a helyi tartományból.

5. táblázat: A kiválasztott parancsok **domainlessgroups** *tulajdonság által befolyásolt eredményei (Folytatás)*

Parancs	Eredmény a domainlessgroups tulajdonság true értéke esetén
lsgroup -R **ldap**|**files**	Felsorolja a megadott tartomány összes csoport attribútumát. Ha nem találja a megadott csoportot a megadott tartományban, akkor a parancs meghiúsul.
lsuser -R **ldap**|**files**	Felsorolja a felhasználó felsorolait, miután az információk összevonásra kerülnek a tartományban lévő összes csoportból, ahol a felhasználó meg van határozva, és a másik tartományból. Ha a felhasználó elsődleges csoportja nincs meghatározva abban a tartományban, ahol a felhasználó meg van határozva, akkor az a másik tartományból lesz feloldva.
mkgroup -R **ldap**|**files**	Létrehoz egy csoportot a megadott tartományban. A csoport létrehozása után hozzárendelheti az (akár LDAP, akár helyi) felhasználót a csoporthoz az adott tartomány csoport adatbázisában. A felhasználót akár LDAP, akár helyi csoporthoz is hozzáadhatja.
mkuser -R **ldap**|**files**	Létrehoz egy felhasználót a megadott tartományban. Ha a másik tartományban meghatározott csoportok meg vannak adva, akkor azok a csoportok szintén frissítve lesznek a felhasználói információkkal.
rmgroup -R **ldap**|**files**	Törli a megadott csoportot a megadott tartományból. Ha a csoport bármely tartományban meghatározott bármely felhasználó esetén elsődleges csoportként van hozzárendelve, akkor a parancs meghiúsul.
rmuser -R **ldap**|**files**	Törli a megadott felhasználót a megadott tartományból. El is távolítja a felhasználót minden olyan csoportból, amely a másik tartományban van meghatározva, és melynek ez a felhasználó a tagja.

Kapcsolódó fogalmak:

“LDAP hitelesítési modul” oldalszám: 150
Az LDAP biztonsági alrendszert az LDAP hitelesítési modul valósítja meg. Alapelvében hasonlít az egyéb betöltési modulokhoz (például NIS, DCE és KRB5). A betöltési modulok az /usr/lib/security/methods.cfg fájlban vannak definiálva.

Kapcsolódó tájékoztatás:

chgroup parancs

chuser parancs

login parancs

lsgroup parancs

lsuser parancs

mkgroup parancs

mkuser parancs

rmgroup parancs

rmuser parancs

su parancs

Jelszavak

A leggyakrabban alkalmazott támadási módszer a jelszavak kitalálása. Ennek megfelelően rendkívüli fontossággal bír a jelszó korlátozási irányelv betartatása és megfigyelése.

Az AIX számos mechanizmust biztosít erősebb jelszó házirendek alkalmazásához, például:

- A jelszó módosítása előtt és után kivárandó minimális és maximális időszak megadása
- A jelszavak minimális hossza
- A jelszavak választásakor használható alfabetikus karakterek minimális száma

Jó jelszavak kialakítása:

A jó jelszavak hatékony első védelmi vonalat jelentenek a rendszerbe jogosulatlanul belépni szándékozókkal szemben:

A jelszavak akkor hatékonyak, ha:

- Nagybetűs és kisbetűs karaktereket is tartalmaznak
- Alfabetikus, numerikus és központozási karakterek keverékéből állnak. Emellett tartalmazhatnak speciális karaktereket, például: ~!@#$%^&*()-_=+[]{}|\;:'",.<>?/<space>
- Sehol sincsenek feljegyezve
- Legalább 7, legfeljebb PW_PASSLEN karakter hosszúak az /etc/security/passwd fájl használatakor. (Nyilvántartásokat használó hitelesítési megvalósítások, például az LDAP lehetővé teszi hosszabb jelszavak használatát is.)
- Nem bírnak jelentéssel, és nem találhatók meg szótárakban
- Nem billentyűzet mintára épülnek, mint például az *asdf*
- Visszafelé olvasva sem alkotnak értelmes szavakat vagy ismert mintákat
- Nem tartalmaznak semmilyen személyes vagy családi információt
- Nem követi az előző jelszavak mintáját
- Viszonylag gyorsan beírható úgy, hogy ne lehessen lelesni a billentyűzetről

Ezen mechanizmusokon felül szigorúbb szabályok is bevezethetők, például a jelszavak nem tartalmazhatnak szabványos UNIX szavakat, amelyeket esetleg ki lehetne találni. Ez a szolgáltatás a szótárlistát használja, amelyhez először telepíteni kell a bos.data és bos.txt fájlkészleteket.

A korábban megadott szótárlista megvalósításához módosítsa az /etc/security/users fájl következő sorát:

```
dictionlist = /usr/share/dict/words
```

A /usr/share/dict/words fájl a szótárlista segítségével megakadályozható, hogy a szabványos UNIX szavakat jelszavakként használják.

A /etc/passwd fájl használata:

Történelmi okokból az /etc/passwd fájl tárolja a rendszerhez hozzáféréssel rendelkező bejegyzett felhasználókat.

Az /etc/passwd fájl egy kettőspontokkal elválasztott fájl, amely a következő információkat tartalmazza:

- Felhasználó neve
- Titkosított jelszó
- Felhasználói azonosító száma (UID)
- Felhasználó csoportazonosító száma (GID)
- Felhasználó teljes neve (GECOS)
- Felhasználó saját könyvtára
- Bejelentkezési héj

Egy példa az /etc/passwd fájlra:

```
root:!:0:0::/:/usr/bin/ksh
daemon:!:1:1::/etc:
bin:!:2:2::/bin:
sys:!:3:3::/usr/sys:
adm:!:4:4::/var/adm:
uucp:!:5:5::/usr/lib/uucp:
guest:!:100:100::/home/guest:
nobody:!:4294967294:4294967294::/:
lpd:!:9:4294967294::/:
lp:*:11:11::/var/spool/lp:/bin/false
```

```
invscout:*:200:1::/var/adm/invscout:/usr/bin/ksh
nuucp:*:6:5:uucp login user:/var/spool/uucppublic:/usr/sbin/uucp/uucico
paul:!:201:1::/home/paul:/usr/bin/ksh
jdoe:*:202:1:John Doe:/home/jdoe:/usr/bin/ksh
```

Az AIX nem az /etc/password fájlban tárolja a titkosított jelszókat, ahogyan a UNIX rendszerek szokták, hanem alapértelmezésben az /etc/security/password[1] fájlban, amely csak a root felhasználó által olvasható. Az /etc/passwd fájlba írt jelszó csak azt jelzi az AIX számára, hogy az azonosítóhoz tartozik-e jelszó, vagy a fiók le van tiltva.

Az /etc/passwd fájl tulajdonosa a root felhasználó, és minden felhasználó számára olvashatónak kell lennie, de csak a root felhasználó írhatja, vagyis az -rw-r--r-- engedélyekkel kell rendelkeznie. Ha egy felhasználói azonosító rendelkezik jelszóval, akkor a jelszó mezőben egy ! (felkiáltójel) áll. Ha a felhasználói azonosító nem rendelkezik jelszóval, akkor a jelszó mezőben egy * (csillag) található. A titkosított jelszavak az /etc/security/passwd fájlban vannak. Az alábbi példa bemutatja az /etc/security/passwd fájl utolsó négy bejegyzését a fenti /etc/passwd fájl alapján.

```
guest:
        password = *

nobody:
        password = *

lpd:
        password = *

paul:
        password = eacVScDKri4s6
        lastupdate = 1026394230
        flags = ADMCHG
```

A jdoe nem rendelkezik bejegyzéssel az /etc/security/passwd fájlban, mivel számára az /etc/passwd fájl nem ad meg jelszót.

A /etc/passwd fájl konzisztenciája a **pwdck** parancs segítségével ellenőrizhető. A **pwdck** parancs az összes felhasználó vagy a kijelölt felhasználó meghatározásainak összehasonlításával ellenőrzi a jelszó információk helyességét a felhasználói adatbázis fájlokban.

A /etc/passwd fájl és a hálózati környezetek használata:

Hagyományos hálózatos környezetben a felhasználónak egy fiókkal kell rendelkeznie minden rendszeren a rendszer elérése érdekében.

Ez általában azt jelentette, hogy a felhasználó minden egyes rendszer /etc/passwd fájljában rendelkezett egy bejegyzéssel. Az osztott környezetekben azonban nincs egyszerű módszer az /etc/passwd fájlok rendszerek közötti összehangolására. A probléma megoldása érdekében számos olyan módszer létezik, amely elérhetővé teszi az /etc/passwd fájlban tárolt információkat a hálózaton. Ilyen például a Hálózati információs rendszer (NIS).

Felhasználói nevek és jelszavak elrejtése:

Magasabb szintű biztonság elérése érdekében győződjön meg róla, hogy a felhasználói azonosítók és jelszavak nem látszanak a rendszerben.

A .netrc fájlok felhasználói azonosítókat és jelszavakat tartalmaznak. A fájlt nem védi kódolás vagy titkosítás, vagyis tartalmaz sima szövegként megtekinthető. Ezen fájlok megkereséséhez futtassa a következő parancsot:

```
# find `awk -F: '{print $6}' /etc/passwd` -name .netrc -ls
```

A megtalált fájlokat törölje le. A jelszavak mentésére a Kerberos beállítása hatékonyabb megoldást jelent. A Kerberos szolgáltatásról további információkat a “Kerberos” oldalszám: 280 szakaszban talál.

1. /etc/security/password

Ajánlott jelszóbeállítások megadása:

A jelszavak megfelelő kezelése csak a felhasználók felvilágosításával érhető el. További biztonság nyújtása érdekében az operációs rendszer biztosít néhány beállítható jelszó korlátozást. Ezek lehetővé teszik az adminisztrátornak a jelszavak korlátozását és rendszeres cseréjét.

A jelszó beállítások és a kiterjesztett felhasználói attribútumok az /etc/security/user fájlban találhatók. Ez a fájl egy ASCII fájlt, benne a felhasználók attribútumait tartalmazó szakaszokkal. A korlátozásokat a rendszer minden egyes alkalommal foganatosítja, amikor egy felhasználó új jelszót kap. Minden jelszó korlátozás felhasználónként érvényesül. A korlátozásoknak az /etc/security/user fájl alapértelmezett szakaszában tárolásával azonos korlátozások foganatosíthatók minden felhasználóra. A jelszavak biztonságának fenntartásához minden jelszót hasonlóan kell védeni.

Az adminisztrátorok kibővíthetik a jelszó korlátozásokat is. Az /etc/security/user fájl **pwdchecks** attribútuma segítségével az adminisztrátor új szubrutinokat (vagy *metódusok*) adhat a jelszókorlátozó kódhoz. Vagyis az adott hely házirendje hozzáadható az operációs rendszerhez. További információk: “Jelszókorlátozások kiterjesztése” oldalszám: 69.

A jelszó korlátozásokat érzéssel kell alkalmazni. A túlzóan megszorító beállítások, amelyek igen nehézzé teszik a jelszó kitalálását, nehézzé teszik a jelszó megjegyzését is, amely esetben elképzelhető, hogy sokan feljegyzést készítenek a jelszóról, így a szigoró korlátozások fordítva sülhetnek el. A jelszó biztonság végső soron a felhasználók kezében van. A legjobb gyakorlat ennek megfelelően néhány egyszerű jelszó korlátozás, kiegészítve néhány jól meghatározott irányvonallal és esetenként egy megfigyeléssel a jelszavak egyediségének ellenőrzéséhez.

A következő táblázat megadja az /etc/security/user fájl néhány jelszavakkal kapcsolatos biztonsági attribútumának ajánlott értékét.

6. táblázat: Felhasználói jelszavak biztonsági attribútumainak ajánlott értékei.

Attribútum	Leírás	Ajánlott érték	Alapértelmezett érték	Maximális érték
dictionlist	Ellenőrzi, hogy a jelszavak nem tartalmaznak szabványos UNIX szavakat.	**/usr/share/dict/words**	n/a	n/a
histexpire	Hetek száma a jelszó ismételt felhasználhatósága előtt.	26	0	260*
histsize	Megengedett jelszó ismétlések száma.	20	0	50
maxage	Hetek maximális száma, mielőtt a jelszót cserélni kell.	8	0	52
maxexpired	A *maxage* utáni hetek maximális száma, amikor a felhasználó még le tudja cserélni a jelszavát. (A root kivételével.)	2	-1	52
maxrepeats	A jelszavakban ismételhető karakterek maximális száma.	2	8	8

6. táblázat: Felhasználói jelszavak biztonsági attribútumainak ajánlott értékei. (Folytatás)

Attribútum	Leírás	Ajánlott érték	Alapértelmezett érték	Maximális érték
minage	A hetek minimális száma, mielőtt a jelszót módosítani lehetne. Ennek nullától eltérő értéket kell megadnia, kivéve azt az esetet, ha az adminisztrátorok mindig könnyen elérhetők a jelszavak alaphelyzetbe állításához egy-egy frissen módosított jelszó nyilvánosságra kerülése esetén.	0	0	52
minalpha	A jelszavakban használandó alfabetikus karakterek minimális száma.	2	0	PW_PASSLEN**
mindiff	A jelszóban tartalmazott egyedi karakterek minimális száma.	4	0	PW_PASSLEN**
minlen	Jelszavak minimális hossza.	6 (root felhasználónál 8)	0	PW_PASSLEN**
minother	A jelszavakban használandó nem alfabetikus karakterek minimális száma.	2	0	PW_PASSLEN**
pwdwarntime	A rendszer ennyi nappal hamarabb küld figyelmeztetést a jelszó módosításának szükségességéről.	5	n/a	n/a
pwdchecks	Ez a bejegyzés használható a **passwd** parancs egyéni kóddal való kiterjesztéséhez a jelszó minőségének további ellenőrzése érdekében.	További információk: "Jelszókorlátozások kiterjesztése" oldalszám: 69.	n/a	n/a

* Maximum 50 jelszó kerül megtartásra.

** A PW_PASSLEN a userpw.h fájlban van megadva

Ha a szöveg feldolgozás telepítve van a rendszeren, akkor az adminisztrátor az /usr/share/dict/words fájlt használhatja **dictionlist** szótárfájlként. Ilyen esetekben az adminisztrátor beállíthatja a **minother** attribútumot 0 értékre, mivel a szótárfájl legtöbb szava nem tartalmaz olyan karaktereket, amelyek a **minother** attribútumkategóriájába esnek, így a **minother** attribútum 1-re vagy annál nagyobbra állítása a szótárfájl legnagyobb részét feleslegessé tenné.

A rendszeren alkalmazott minimális jelszóhosszt a **minlen** értéke vagy a **minalpha** és **minother** attribútumok értékeinek összege közül a nagyobb érték határozza meg.

A jelszó maximális hossza a **PW_PASSLEN** attribútumban megadott karakterszám. A tárolt jelszóérték előállításakor használt karakterek száma a rendszeren használt jelszóalgoritmustól függ. A jelszóalgoritmusok az /etc/security/pwdalg.cfg fájlban vannak megadva, és a használandó alapértelmezett jelszóalgoritmus az /etc/security/login.cfg fájl **pwd_algorithm** attribútumával állítható be. A **minalpha** és a **minother** attribútum

értékének összege nem lehet nagyobb a **PW_PASSLEN** attribútum értékénél. Ha a **minalpha** és a **minother** attribútum értékének összege nagyobb mint a **PW_PASSLEN** attribútum, akkor a **minother** attribútum értékét a rendszer **PW_PASSLEN** - **minalpha** értékre állítja.

Ha a **histexpire** és a **histsize** attribútum is be van állítva, akkor a rendszer megkísérli kielégíteni mindkét feltételt, és a felhasználónkénti 50-es korlátozásig megtartja a szükséges számú jelszót. Az üres jelszavak nem kerülnek megtartásra.

A /etc/security/user fájl módosításával tetszőleges alapértelmezéseket adhat meg a felhasználói jelszavakra vonatkozóan. Ennek alternatívájaként az attribútumok értékeit a **chuser** paranccsal is módosíthatja.

A fájlon használható további parancsok: **mkuser**, **lsuser** és **rmuser**. Az **mkuser** parancs minden egyes új felhasználó számára létrehoz egy bejegyzést az /etc/security/user fájlban, és inicializálja az attribútumait az /usr/lib/security/mkuser.default fájlban megadott attribútumok alapján. Az attribútumok és ezek értékeinek megjelenítésére használja az **lsuser** parancsot. Egy felhasználó eltávolításához használja az **rmuser** parancsot.

A 8-nál több karaktert tartalmazó jelszavak és a betölthető jelszó algoritmus támogatása:

A számítógéphardver terén elért jelenlegi fejlődés a hagyományos UNIX jelszótitkosítást támadhatóvá teszi a próbálgatáson alapuló jelszómegfejtési támadásokkal szemben. A kriptográfiailag gyenge algoritmus még az erős jelszavak visszafejtését is lehetővé teheti. Az AIX támogatja a Betölthető jelszó algoritmust (LPA), amely biztonságos jelszókivonatolási mechanizmusokat biztosít.

Hagyományos jelszó crypt függvény:

A szabványos AIX hitelesítési mechanizmus egyirányú **crypt** nevű kivonatkészítési függvényt használ a felhasználók hitelesítéséhez. A **crypt** függvény módosított DES algoritmus. A rögzített adattömb egyirányú titkosítását hajtja végre a biztosított jelszóval és variációs értékkel (Salt).

A **crypt** függvény csak a jelszó karaktersorozat első nyolc karakterét használja. A felhasználói jelszót a rendszer nyolc karakterre csonkítja. Ha a jelszó kevesebb mint nyolc karaktert tartalmaz, akkor a jobb oldalához nulla bitek kerülnek hozzáfűzésre. Az 56 bites DES kulcs a karakterek 7 bitjének felhasználásával kerül származtatásra.

A variációs érték az "**A**-**Z**", "**a**-**z**","**0**-**9**","**.**"(pont) és "/" karakterkészletből választott kétkarakteres karaktersorozat (a variációs érték 12 bitje összezavarja a DES algoritmust). A variációs érték megváltoztatja a kivonatkészítési algoritmust, így ugyanaz a sima szövegű jelszú 4 096 lehetséges jelszótitkosítást állíthat elő. A DES algoritmus módosítása, a DES E-Box kimenet i. és i+24. bitjének cseréje, ha az i. bit be van állítva a variációs értékben, ezt éri el, és ezáltal a DES titkosítási hardver nem használható a jelszókitalálásra.

A 64 bites összes-bit-nulla blokk 25-ször kerül titkosításra a DES kulccsal. A végső kimenet a 12 bites variációs érték hozzáfűzve a titkosított 64 bites értékhez. Az eredményül kapott 76 bites érték rögzítésre kerül 13 nyomtatható ASCII karakterben, base64 formában.

Jelszó-kivonatolási algoritmusok:

A kivonatolási algoritmusok, mint például az MD5, feltörése nehezebb, mint a **crypt** függvényé. Ez erős mechanizmust biztosít a próbálgatáson alapuló jelszókitalálásos támadásokkal szemben. Mivel a teljes jelszó felhasználásra kerül a kivonat előállításához, nincs jelszóhosz-korlátozás, ha jelszókivonatolási algoritmusokat használ a jelszó titkosításához.

Betölthető jelszó algoritmus:

Az AIX 6.1 és újabb megvalósított egy LPA mechanizmust, amely egyszerűen tud új jelszótitkosítási algoritmusokat bevezetni.

Minden támogatott titkosítási algoritmus LPA betölthető modulként kerül megvalósításra, amely futási időben kerül betöltésre, amikor az algoritmusra szükség van. A támogatott LPA-k és attribútumai az /etc/security/pwdalg.cfg rendszerkonfigurációs fájlban vannak megadva.

Az adminisztrátor beállíthat rendszerszintű jelszótitkosítási mechanizmust, amely adott LPA-t használ a jelszavak titkosításához. A rendszerszintű jelszótitkosítási mechanizmus módosítása után a korábban kiválasztott jelszótitkosítási mechanizmusokkal, mint például a **crypt** függvény, titkosított jelszavak továbbra is támogatottak.

Nyolc karakternél hosszabb jelszavak támogatása:

Az AIX 6.1 és újabb változathoz megvalósított minden LPA támogatja a nyolc karakternél hosszabb jelszavakat. A jelszóhossz-korlátozás a különböző LPA-k esetén eltérő. A támogatott maximális jelszóhossz 255 karakter.

LPA konfigurációs fájl:

Az LPA konfigurációs fájl az /etc/security/pwdalg.cfg. Az a támogatott LPA-k attribútumait meghatározó szakaszfájl.

A következő LPA attribútumok vannak megadva a konfigurációs fájlban:

- Az LPA modul elérési útja
- Az LPA modulnak futási időben átadott elhagyható jelzők

A konfigurációs fájlban megadott LPA attribútumok a **getconfattr** és **setconfattr** felület segítségével elérhetők.

Az /etc/security/pwdalg.cfg fájlban lévő következő példa szakasz egy **ssha256** nevű LPA-t ad meg:

```
ssha256:
    lpa_module = /usr/lib/security/ssha
    lpa_options = algorithm=sha256
```

Rendszerjelszó-algoritmus:

A rendszeradminisztrátor beállíthat egy rendszerszintű jelszóalgoritmust az LPA jelszókivonatolási algoritmusként való kiválasztásával. Egyszerre csak egy aktív rendszerjelszó-algoritmus lehet. A rendszerjelszó-algoritmust az /etc/security/login.cfg fájl **usw** szakaszában lévő **pwd_algorithm** attribútum adja meg.

Az /etc/security/login.cfg fájlban lévő **pwd_algorithm** attribútum érvényes értékei az /etc/security/pwdalg.cfg fájlban megadott szakasznevek. A **pwd_algorithm** attribútum másik érvényes értéke a **crypt**, amely a hagyományos **crypt** titkosításra utal. Ha a **pwd_algorithm** attribútum ki van hagyva a konfigurációs fájlból, akkor alapértelmezett értékként a **crypt** kerül felhasználásra.

A következő példában az /etc/security/login.cfg fájl **ssha256** LPA-t használ rendszerszintű jelszótitkosítási algoritmusként.

```
 ... ...
usw:
   shells = /bin/sh,/bin/bsh,/bin/csh,/bin/ksh,/bin/tsh,/bin/ksh93
   maxlogins = 32767
   logintimeout = 60
   maxroles = 8
   auth_type = STD_AUTH
   pwd_algorithm = ssha256
 ... ...
```

A rendszerjelszó-algoritmus csak az újonnan létrehozott és módosított jelszavakra van hatással. Átállítás után minden további új jelszó vagy jelszómódosítás a rendszerjelszó-algoritmust használja. A rendszerjelszó-algoritmus kiválasztása előtt már meglévő jelszavak, amelyeket a szabványos **crypt** függvény vagy más támogatott LPA modul állított elő, továbbra is működnek a rendszeren. Ezáltal a különböző LPA-k által előállított vegyes jelszavak együtt létezhetnek a rendszeren.

Rendszerjelszó-algoritmus beállítása:

A rendszeradminisztrátor a **chsec** parancs segítségével beállíthatja a rendszerjelszó-algoritmust vagy egy szerkesztő - mint például a **vi** - segítségével kézzel módosíthatja a az /etc/security/login.cfg fájlban lévő **pwd_algorithm** attribútumot.

Ajánlatos a **chsec** paranccsal beállítani a rendszerjelszó-algoritmust, mivel a **chsec** parancs automatikusan ellenőrzi a megadott LPA meghatározását.

A chsec parancs használata

Futtassa a következő parancsot az **smd5** LPA használatához rendszerszintű jelszótitkosításként:

```
chsec -f /etc/security/login.cfg -s usw -a pwd_algorithm=smd5
```

Ha a **chsec** paranccsal módosítja a **pwd_algorithm** attribútumot, akkor a **chsec** parancs megnézi az /etc/security/pwdalg.cfg fájlt a megadott LPA ellenőrzése érdekében. A **chsec** parancs meghiúsul, ha ez az ellenőrzés meghiúsul.

Szerkesztő használata

Ha egy szerkesztő segítségével saját kezűleg módosítja az /etc/security/login.cfg fájlban a **pwd_algorithm** attribútum értékét, akkor győződjön meg róla, hogy a megadott érték az /etc/security/pwdalg.cfg fájlban megadott szakasz neve.

Jelszókorlátozások kiterjesztése:

A jelszó program által a jelszavak elfogadásakor vagy visszautasításakor alkalmazott szabályokat (jelszó összeállítási korlátozásokat) a rendszeradminisztrátorok kibővíthetik az adott környezetre jellemző egyéni korlátozásokkal.

A korlátozások módszerek hozzáadásával terjeszthetők ki, amelyek a jelszavak módosításakor kerülnek meghívásra. A /etc/security/user fájl **pwdchecks** attribútuma a meghívott metódusokat adja meg.

A *AIX Version 6.1 Technical Reference* változatával kezdődően a **pwdrestrict_method** leírását tartalmazza. Ez egy szubrutinfelület, amelyhez a megadott jelszókorlátozási metódusoknak meg kell felelniük. A jelszó összeállítási korlátozások helyes kiterjesztéséhez a rendszeradminisztrátornak ezt a felületet kell használnia jelszó korlátozási módszerek írására. A jelszó összeállítási korlátozások kiterjesztésekor óvatosan kell eljárni. Ezek a kiterjesztések közvetlen hatással vannak a **login**, a **passwd**, a **su** parancsra valamint egyéb programokra. A rendszer biztonsága könnyen alááshető egy rosszindulató vagy hibás kóddal.

Felhasználó hitelesítése

A felhasználó azonosságát az azonosítás és hitelesítés határozza meg.

Minden felhasználónak be kell jelentkeznie a rendszerbe. A felhasználó megadja egy fiók felhasználói nevét és egy jelszót, amennyiben a fiók rendelkezik jelszóval is. (Biztonságos rendszerekben minden felhasználónak rendelkeznie kell jelszóval, vagy tiltottnak kell lennie.) Ha a jelszó megfelelő, akkor a felhasználó bejelentkezik az adott fiókba, vagyis megkapja az adott fiók hozzáférési jogait és privilégiumait. A felhasználói jelszavakat az /etc/passwd és az /etc/security/passwd fájlok tárolják.

A felhasználói meghatározások alapértelmezésben a Fájlok nyilvántartásban találhatók. Ez azt jelenti, hogy a felhasználói fiók információi egyszerű ASCII szöveges fájlokban tárolódnak. A bedolgozó modulok segítségével azonban a felhasználókat más nyilvántartásokban is meg lehet adni. Ha például a felhasználó adminisztrációhoz LDAP bedolgozómodult használ, akkor a felhasználói meghatározások egy LDAP címtárban tárolódnak. Ebben az esetben a felhasználókhoz nem tartozik bejegyzés az /etc/security/user fájlban (ez alól kivételt csak a **SYSTEM** és a **registry** felhasználói attribútum képez). Amikor a felhasználó hitelesítés egy összetett betöltődő modul (például egy hitelesítési- és adatbázis résszel rendelkező betöltődő modul) segítségével történik, akkor az adatbázis rész határozza meg az AIX felhasználói információk adminisztrációjának módját, míg a hitelesítési modul felelős a hitelesítéssel és jelszavakkal

kapcsolatos adminisztrációért. A hitelesítési rész ezenkívül a felhasználói fiók hitelesítéssel kapcsolatos adminisztrációs jellemzőit is kezeli bizonyos betöltődő modul felületek megvalósításával (newuser, getentry, putentry stb).

Ezt a hitelesítési módszert az /etc/security/user file fájlban meghatározott **SYSTEM** és nyilvántartás attribútumok vezérlik. A rendszeradminisztrátor megadhatja az authcontroldomain attribútumot az /etc/security/login.cfg fájlban annak kikényszerítéséhez, hogy a **SYSTEM** és nyilvántartás attribútumok az authcontroldomain tartományból legyenek lekérve. Például az authcontroldomain=LDAP arra kényszeríti a rendszert, hogy az LDAP címtárból kikeresse a felhasználóhoz tartozó **SYSTEM** és nyilvántartás attribútumot, hogy meghatározza a felhasználóhoz használt hitelesítési módszert. Kivételt képeznek ez alól a helyileg meghatározott felhasználók, ahol az authcontroldomain beállítás figyelmen kívül marad, és a **SYSTEM** és a nyilvántartás mindig az /etc/security/user fájlból kerül kikeresésre.

Az authcontroldomain attribútumhoz elfogadható token: fájlok vagy egy szakasz neve a /usr/lib/security/methods.cfg fájlból.

A **SYSTEM** attribútum értékét meghatározott szabályok szerint lehet megadni. Ennek a "nyelvtannak" a segítségével a rendszeradminisztrátorok akár több módszert is kombinálhatnak egy felhasználó hitelesítéséhez. A közismert módszer tokenek: compat, DCE, files és NONE.

Az alapértelmezés a compat. Az alapértelmezett SYSTEM=compat beállítás a helyi hitelesítési adatbázis használatát írja elő. Ha a feloldás sikertelen, akkor a rendszer a Hálózati információs szolgáltatás (NIS) adatbázissal próbálkozik. A files token esetében csak a helyi fájlok használhatóak, míg a SYSTEM=DCE egy DCE hitelesítési folyamatot indít el.

A NONE token kikapcsolja a hitelesítési metódusok használatát. Minden hitelesítés kikapcsolásához a NONE tokennek a felhasználói szakasz SYSTEM és auth1 sorában is szerepelnie kell.

Két vagy több metódus kombinálását az AND és OR logikai operátorokkal írhatja elő. Például: SYSTEM=DCE OR compat . Ez azt jelzi, hogy a felhasználó akkor jelentkezhet be, ha a DCE vagy a helyi hitelesítés (crypt()) ebben a sorrendben sikerrel jár.

Ehhez hasonló módon a rendszeradminisztrátor hitelesítési betöltődő modulok neveit is felhasználhatja a **SYSTEM** attribútumban. Ha például a **SYSTEM** attribútum értéke "SYSTEM=KRB5files OR compat", akkor a AIX hoszt először egy Kerberos hitelesítési folyamatot indít el, és ha az sikertelen, akkor megpróbálkozik a szabványos AIX hitelesítéssel.

A **SYSTEM** és **registry** attribútumok mindig a helyi fájlrendszeren tárolódnak a /etc/security/user fájlban. Ha egy AIX felhasználó meg van adva az LDAP címtárban és a **SYSTEM** és **registry** attribútumok megfelelően vannak beállítva, akkor az /etc/security/user fájlban a felhasználóhoz tartozik egy bejegyzés.

A felhasználó **SYSTEM** és **registry** attribútuma a **chuser** paranccsal változtatható meg.

A **SYSTEM** attribútum elfogadható tokenjei az /usr/lib/security/methods.cfg fájlban adhatók meg.

Megjegyzés: A root felhasználó mindig a helyi rendszer biztonsági fájl szerint kerül hitelesítésre. A root felhasználó **SYSTEM** attribútuma a /etc/security/user fájlban SYSTEM=compat értékre van állítva.

Alternatív hitelesítési módszerek a /etc/security/user fájlban megadott **SYSTEM** attribútummal integrálhatók a rendszerbe. Az Osztott számítási környezet (DCE) például szintén igényel jelszavas hitelesítést, de a jelszavakat az etc/passwd és az /etc/security/passwd fájlokban alkalmazott titkosítási modelltől eltérő módszerrel ellenőrzi. A DCE módszerrel hitelesített felhasználókra vonatkozó szakasz az /etc/security/user fájlban SYSTEM=DCE értékre lehet állítva.

További **SYSTEM** attribútumérték például a **compat**, a **files** és a **NONE**. A compat token akkor használható, ha a névfeloldás (és az azt követő hitelesítés a helyi adatbázist követi, és ha nem található feloldás, akkor próbálkozik meg a rendszer a Hálózati információs szolgáltatás (NIS) adatbázissal. A files token azt adja meg, hogy a hitelesítés során

csak a helyi fájlok kerülnek felhasználásra. Végül a NONE token kikapcsolja a módszer hitelesítést. Minden hitelesítés kikapcsolásához a NONE tokennek a felhasználói szakasz **SYSTEM** és **auth1** sorában is szerepelnie kell.

A **SYSTEM** attribútumhoz további elfogadható tokenek az /usr/lib/security/methods.cfg fájlban adhatók meg.

Megjegyzés: A root felhasználó mindig a helyi rendszer biztonsági fájl szerint kerül hitelesítésre. A root felhasználó **SYSTEM** attribútuma az /etc/security/user fájlban a SYSTEM = "compat" értékre van állítva.

A jelszavak védelméről további információkat az *Operating system and device management* című kiadványban talál.

Bejelentkezési felhasználói azonosítók

A felhasználóval kapcsolatban feljegyzett valamennyi megfigyelési esemény ezzel az azonosítóval kerül címkézésre. A bejelentkezési felhasználói azonosítókról további információkat az *Operating system and device management* című kiadványban talál.

Hitelesítés betöltő modulok által támogatott felhasználói és csoport attribútumok

Az azonosítást és a hitelesítést felhasználói és csoport attribútumok egy csoportja határozza meg az AIX rendszerekben.

Az alábbi táblázat a legtöbb ilyen felhasználói és csoport attribútumot mutatja be, és jelzi az attribútumok támogatottságát a különböző betöltési moduloknál. A táblázat sorai egy-egy attribútumnak, oszlopai pedig egy-egy betöltési modulnak felelnek meg. A betöltési modul által támogatott attribútumot Igen jelzi a betöltési modul oszlopában.

Megjegyzés: A PKI és a Kerberos csak hitelesítési modulok, így ezeket adatbázis modellel kell kombinálni (például LOCAL vagy LDAP). Ezek további (kiterjesztett) attribútumokat is támogatnak a LOCAL vagy LDAP által biztosított attribútumokon kívül. A jelzések ezeknél a moduloknál csak ezeknél a kiterjesztett attribútumoknál szerepelnek még akkor is, ha a többi attribútum funkcionalitását LOCAL vagy LDAP használatával is el lehet érni.

7. táblázat: Felhasználói attribútumok és hitelesítés betöltési modul támogatás

Felhasználói attribútum	Helyi	NIS	LDAP	PKI	Kerberos
account_locked	Igen	Nem	Igen	Nem	Nem
admgroups	Igen	Nem	Igen	Nem	Nem
admin	Igen	Nem	Igen	Nem	Nem
auditclasses	Igen	Nem	Igen	Nem	Nem
auth_cert	Nem	Nem	Nem	Igen	Nem
auth_domain	Igen	Nem	Igen	Nem	Nem
auth_name	Igen	Nem	Igen	Nem	Nem
auth1 **Megjegyzés:** Az **auth1** attribútum elavult és nem használható.	Igen	Nem	Igen	Nem	Nem
auth2 **Megjegyzés:** Az **auth2** attribútum elavult és nem használható.	Igen	Nem	Igen	Nem	Nem
capabilities	Igen	Nem	Igen	Nem	Nem
core	Igen	Nem	Igen	Nem	Nem
core_compress	Igen	Nem	Nem	Nem	Nem
core_hard	Igen	Nem	Igen	Nem	Nem
core_naming	Igen	Nem	Nem	Nem	Nem
core_path	Igen	Nem	Nem	Nem	Nem
core_pathname	Igen	Nem	Nem	Nem	Nem
cpu	Igen	Nem	Igen	Nem	Nem

7. táblázat: Felhasználói attribútumok és hitelesítés betöltési modul támogatás (Folytatás)

Felhasználói attribútum	Helyi	NIS	LDAP	PKI	Kerberos
daemon	Igen	Nem	Igen	Nem	Nem
data	Igen	Nem	Igen	Nem	Nem
data_hard	Igen	Nem	Igen	Nem	Nem
dce_export	Igen	Nem	Igen	Nem	Nem
dictionlist	Igen	Nem	Igen	Nem	Nem
expires	Igen	Nem	Igen	Nem	Igen
flags	Igen	Nem	Igen	Nem	Igen
fsize	Igen	Nem	Igen	Nem	Nem
fsize_hard	Igen	Nem	Igen	Nem	Nem
funcmode	Igen	Nem	Igen	Nem	Nem
gecos	Igen	Igen	Igen	Nem	Nem
groups	Igen	Igen	Igen	Nem	Nem
groupsids	Igen	Igen	Igen	Nem	Nem
histexpire	Igen	Nem	Igen	Nem	Nem
home	Igen	Igen	Igen	Nem	Nem
host_last_login	Igen	Nem	Igen	Nem	Nem
host_last_unsuccessful_login	Igen	Igen	Igen	Nem	Nem
hostsallowedlogin	Igen	Nem	Igen	Nem	Nem
hostsdeniedlogin	Igen	Nem	Igen	Nem	Nem
id	Igen	Igen	Igen	Nem	Nem
krb5_attributes	Nem	Nem	Nem	Nem	Igen
krb5_kvno	Nem	Nem	Nem	Nem	Igen
krb5_last_pwd_change	Nem	Nem	Nem	Nem	Igen
krb5_max_renewable_life	Nem	Nem	Nem	Nem	Igen
krb5_mknvo	Nem	Nem	Nem	Nem	Igen
krb5_mod_date	Nem	Nem	Nem	Nem	Igen
krb5_mod_name	Nem	Nem	Nem	Nem	Igen
krb5_names	Nem	Nem	Nem	Nem	Igen
krb5_principal	Nem	Nem	Nem	Nem	Igen
krb5_principal_name	Nem	Nem	Nem	Nem	Igen
krb5_realm	Nem	Nem	Nem	Nem	Igen
lastupdate	Igen	Igen	Igen	Nem	Nem
login	Igen	Nem	Igen	Nem	Nem
loginretries	Igen	Nem	Igen	Nem	Nem
logintimes	Igen	Nem	Igen	Nem	Nem
maxage	Igen	Igen	Igen	Nem	Igen
maxexpired	Igen	Igen	Igen	Nem	Nem
maxrepeats	Igen	Nem	Igen	Nem	Nem
maxulogs	Igen	Nem	Igen	Nem	Nem
minage	Igen	Igen	Igen	Nem	Nem
minalpha	Igen	Nem	Igen	Nem	Nem
mindiff	Igen	Nem	Igen	Nem	Nem
mindigit	Igen	Nem	Igen	Nem	Nem
minlen	Igen	Nem	Igen	Nem	Nem
minloweralpha	Igen	Nem	Igen	Nem	Nem

7. táblázat: Felhasználói attribútumok és hitelesítés betöltési modul támogatás (Folytatás)

Felhasználói attribútum	Helyi	NIS	LDAP	PKI	Kerberos
minother	Igen	Nem	Igen	Nem	Nem
minspecialchar	Igen	Nem	Igen	Nem	Nem
minupperalpha	Igen	Nem	Igen	Nem	Nem
nofiles	Igen	Nem	Igen	Nem	Nem
nofiles_hard	Igen	Nem	Igen	Nem	Nem
jelszó	Igen	Igen	Igen	Nem	Nem
pgid	Igen	Igen	Nem	Nem	Nem
pgrp	Igen	Igen	Igen	Nem	Nem
projects	Igen	Nem	Igen	Nem	Nem
pwdchecks	Igen	Nem	Igen	Nem	Nem
pwdwarntime	Igen	Nem	Igen	Nem	Nem
rcmds	Igen	Nem	Igen	Nem	Nem
registry	Igen	Nem	Nem	Nem	Nem
rlogin	Igen	Nem	Igen	Nem	Nem
roles	Igen	Nem	Igen	Nem	Nem
rss	Igen	Nem	Igen	Nem	Nem
rss_hard	Igen	Nem	Igen	Nem	Nem
screens	Igen	Nem	Igen	Nem	Nem
shell	Igen	Igen	Igen	Nem	Nem
spassword	Igen	Igen	Igen	Nem	Nem
stack	Igen	Nem	Igen	Nem	Nem
stack_hard	Igen	Nem	Igen	Nem	Nem
su	Igen	Nem	Igen	Nem	Nem
sugroups	Igen	Nem	Igen	Nem	Nem
sysenv	Igen	Nem	Igen	Nem	Nem
SYSTEM	Igen	Nem	Nem	Nem	Nem
time_last_login	Igen	Nem	Igen	Nem	Nem
time_last_unsuccessful_login	Igen	Nem	Igen	Nem	Nem
tpath	Igen	Nem	Igen	Nem	Nem
tty_last_login	Igen	Nem	Igen	Nem	Nem
tty_last_unsuccessful_login	Igen	Nem	Igen	Nem	Nem
ttys	Igen	Nem	Igen	Nem	Nem
umask	Igen	Nem	Igen	Nem	Nem
unsuccessful_login_count	Igen	Nem	Igen	Nem	Nem
unsuccessful_login_times	Igen	Nem	Igen	Nem	Nem
usrenv	Igen	Nem	Igen	Nem	Nem

8. táblázat: Csoport attribútumok és hitelesítés betöltési modul támogatás

Felhasználói attribútum	Helyi	NIS	LDAP	PKI	Kerberos
admin	Igen	Nem	Igen	Nem	Nem
adms	Igen	Nem	Igen	Nem	Nem
dce_export	Igen	Nem	Igen	Nem	Nem
id	Igen	Igen	Igen	Nem	Nem
primary	Igen	Nem	Igen	Nem	Nem
projects	Igen	Nem	Igen	Nem	Nem
screens	Igen	Nem	Igen	Nem	Nem
users	Igen	Igen	Igen	Nem	Nem

Lemezkvótarendszer áttekintése

A lemezkvótarendszer segítségével az adminisztrátorok meghatározhatják, hogy legfeljebb hány fájl és hány adatblokk foglalható le egy adott felhasználó vagy csoport számára.

Lemezkvótarendszer alapelve:

A lemezkvóta rendszer a Berkeley lemezkvóta rendszeren alapszik, és képes hatékonyan felügyelni a lemezhasználatot. A kvótarendszer definiálható egyedi felhasználókra vagy csoportokra. A kvótarendszer az egyes naplózott fájlrendszerekre (JFS és JFS2) külön-külön lehet karbantartani.

A lemezkvóta rendszer a korlátokat az alábbi paraméterek alapján alakítja ki, amelyek JFS fájlrendszereknél az **edquota**, JFS2 fájlrendszereknél pedig a **j2edlimit** paranccsal módosíthatók:

- Felhasználó vagy csoport puha korlátai
- Felhasználó vagy csoport kemény korlátai
- Kvóta türelmi idő

A *puha korlát* adja meg a felhasználó vagy csoport által a normál műveletek során használható 1 KB-os lemezblokkok vagy fájlok számát. A *kemény korlát* a felhasználó által felhasználható maximális lemezblokkok vagy fájlok mennyiségét adja meg a megadott lemezkvótákon belül. A *kvóta türelmi idő* lehetővé teszi a felhasználó számára, hogy egy rövid időre (az alapértelmezett érték egy hét) túllépje a puha korlátot. Ha a felhasználó a megadott idő alatt nem csökkenti a lemezhasználatot a puha korláton belülre, akkor a rendszer a puha korlátot maximálisan engedélyezett kiosztásként kezeli, és nem oszt ki további területet a felhasználó számára. A felhasználó ügy szüntetheti meg ezt a helyzetet, hogy fájlok eltávolításával a puha korláton belülre korlátozza a lemezhasználatot.

A lemezkvóta rendszer a felhasználói- és csoportkvótákat a quota.user és quota.group fájlokban követi nyomon a kvótákat használó fájlrendszer gyökérkönyvtárában. A fájlok a **quotacheck** és az **edquota** parancsokkal hozhatók létre, és a kvóta parancsokkal olvashatók.

Kvótatúllépési helyzetek helyreállítása:

A kvótatúllépés után a rendszert a fájlrendszer használatának csökkentésével helyreállíthatja.

Fájlhasználat csökkentésének módjai, ha túllépte a kvóta korlátokat:

- Állítsa le azt az aktuális folyamatot, amely miatt a fájlrendszer elérte a korlátot, távolítsa el a többlet fájlokat, hozza a korlátot a kvóta alá, majd futtassa ismét a sikertelen programot.
- A szerkesztőt használ - például vi-t -, akkor a héj vezérlő jelsorozattal ellenőrizze a fájlhelyet, távolítsa el a többlet fájlokat, és térjen vissza a szerkesztett fájl elvesztése nélkül. Vagy ha C vagy Korn héjt használ, akkor függessze fel a szerkesztőt a Ctrl-Z billentyűkombinációval, adja ki a fájlrendszer parancsokat, majd térjen vissza az **fg** (előtér) paranccsal.
- Ideiglenesen írja a fájlt egy olyan fájlrendszerre, ahol a kvóta határokat még nem haladta meg, törölje a többlet fájlokat, majd hozza vissza a fájlt a megfelelő fájlrendszerbe.

A lemezkvótarendszer beállítása:

Általában csak a felhasználói saját könyvtárakat és fájlokat tartalmazó fájlrendszereknek van szükségük lemezkvótára.

A lemezkvóta rendszer megvalósításakor tartsa szem előtt az alábbi körülményeket:

- A rendszerben a lemezterület mérete korlátozott.
- Nagyobb fájlrendszer biztonságra van szükség.
- A lemezhasználat igen nagy, mint például számos nagy egyetemen.

Ha ezek a körülmények nem vonatkoznak az adott környezetre, akkor nincs szükség lemezhasználati korlátok bevezetésére a lemezkvóta rendszer megvalósításával.

A lemezkvóta rendszer csak a naplózott fájlrendszerrel használható.

Megjegyzés: Ne hozzon létre lemezkvótákat a /tmp fájlrendszerre.

Lemezkvótarendszer beállításához tegye a következőket:

1. Jelentkezzen be root jogosultsággal.
2. Határozza meg, hogy mely fájlrendszereken van szükség kvótákra.

   Megjegyzés: Mivel számos szerkesztő és rendszer segédprogram hoz létre ideiglenes fájlokat a /tmp fájlrendszerben, ezért ennek kvótáktól mentesnek kell lennie.
3. A **chfs** parancs segítségével veheti fel a **userquota** és **groupquota** kvótakonfigurációs attribútumokat a /etc/filesystems fájlba. Az alábbi példa a **chfs** paranccsal engedélyezi a felhasználói kvótákat a /home fájlrendszeren:

   ```
   chfs -a "quota = userquota" /home
   ```

   Ha a felhasználói- és a csoportkvótákat is engedélyezni szeretné a /home fájlrendszeren, akkor írja be a következő parancsot:

   ```
   chfs -a "quota = userquota,groupquota" /home
   ```

   Az /etc/filesystems fájl megfelelő bejegyzése a következőképpen néz ki:

   ```
   /home:
   dev        = /dev/hd1
   vfs        = jfs
   log        = /dev/hd8
   mount      = true
   check      = true
   quota      = userquota,groupquota
   options    = rw
   ```
4. Alternatív lemezkvóta fájlnevek is megadhatók. A quota.user és a quota.group alapértelmezett fájlnevek, és a kvótákat használó fájlrendszer gyökérkönyvtárában találhatók. Más neveket és könyvtárakat is megadhat a kvóta fájloknak a **userquota** és a **groupquota** attribútumokkal az /etc/filesystems fájlban.

   Az alábbi példában a **chfs** parancs felhasználói- és csoportkvótákat állapít meg a /home fájlrendszerre, a myquota.user és a myquota.group kvótafájlokkal:

   ```
   chfs -a "userquota = /home/myquota.user" -a "groupquota = /home
                  /myquota.group" /home
   ```

   Az /etc/filesystems fájl megfelelő bejegyzése a következőképpen néz ki:

   ```
   /home:
   dev        = /dev/hd1
   vfs        = jfs
   log        = /dev/hd8
   mount      = true
   check      = true
   ```

```
quota      = userquota,groupquota
userquota  = /home/myquota.user
groupquota = /home/myquota.group
options    = rw
```

5. Ha a megadott fájlrendszerek korábban nem kerültek felkapcsolásra, akkor kapcsolja fel azokat.
6. Állítsa be a kívánt kvótakorlátokat az egyes felhasználók és csoportok számára. Az **edquota** paranccsal hozzon létre puha és kemény korlátokat minden felhasználó és csoport számára a megengedett lemezterületre és a fájlok maximális számára.

   Az alábbi példa a *davec* felhasználó kvóta korlátait mutatja:

   ```
   davec felhasználó kvótái:
   /home: használt blokkok: 30, korlátok (soft = 100, hard = 150)
                  használt inode-ok: 73, korlátok (soft = 200, hard = 250)
   ```

   A felhasználó 30 KB-ot használ a maximális 100 KB lemezterületből. A maximális 200 fájlból *davec* 73-at hozott létre. A felhasználónak 50 KB-os puffer lemezterülete és 50 fájlja van, amelyek ideiglenes tárolóba oszthatók ki.

   Ha több felhasználó számára hoz létre lemezkvótákat, akkor az **edquota** parancs **-p** kapcsolójával lemásolhatja a felhasználói kvótákat egy másik felhasználó számára.

   Ha a *davec* felhasználó kvótáit szeretné megadni a *nanc* felhasználó számára, akkor írja be a következő parancsot:

   ```
   edquota -p davec nanc
   ```

7. A **quotaon** paranccsal engedélyezze a kvótarendszert. A **quotaon** parancs engedélyezi a kvótákat a megadott fájlrendszerre, vagy az összes kvótával rendelkező fájlrendszerre (az /etc/filesystems fájl alapján), ha az **-a** kapcsolóval kerül megadásra.
8. A **quotacheck** paranccsal ellenőrizheti a kvótafájlok és a ténylegesen felhasznált lemezterület konzisztenciáját.

   Megjegyzés: Ezt hajtsa végre minden olyan alkalommal, amikor először engedélyez kvótákat egy fájlrendszeren, illetve a rendszer újraindítása után. A **quotacheck** parancs futása JFS fájlrendszeren tovább tart, mint ugyanolyan méretű JFS2 fájlrendszeren. Ha az újraindítás előtt minden engedélyezve vannak kvóták, akkor az újraindítás során a fájlrendszeren nem kell futtatni a **quotacheck** parancsot.

   Az ellenőrzés engedélyezéséhez és a kvóta bekapcsolásához a rendszerindítás során adja hozzá a következő sorokat az /etc/rc fájlhoz:

   ```
   echo " Enabling filesystem quotas "
   /usr/sbin/quotacheck -a
   /usr/sbin/quotaon -a
   ```

Csoportok megengedett száma

Beállíthatja és lekérheti az AIX 7.1 Csoportok megengedett száma értékét. Ez meghatározza, hogy a felhasználók hány csoportnak lehetnek tagjai.

A Csoportok megengedett száma alapértelmezésben 128. A 128 és a 2048 közötti tartományban változtatható. A Csoportok megengedett száma értéket a v_ngroups_allowed rendszerkonfigurációs paraméterrel lehet megadni a sys0 eszközhöz. Módosíthatja vagy lekérheti a v_ngroups_allowed paraméter értékét a kernelről vagy az ODM adatbázisból. A kernelben lévő paraméterértéket a futás közben használja a rendszer. Az ODM adatbázisban lévő érték a rendszer újraindítása után lép érvénybe.

Csoportok megengedett számának lekérése az ODM adatbázisból: A v_ngroups_allowed paraméter parancsok vagy szubrutinok használatával kérhető le. Az **lsattr** parancs kiadásával kérheti le a v_ngroups_allowed paramétert az ODM adatbázisból.

Az **lsattr** parancs a v_ngroups_allowed paramétert ngroups_allowed attribútumként jeleníti meg. A következő példa bemutatja, hogy az **lsattr** parancs segítségével hogyan kérhető le az ngroups_allowed attribútum:

```
$ lsattr -El sys0
SW_dist_intr    false           Enable SW distribution of interrupts             True
autorestart     true            Automatically REBOOT system after a crash        True
boottype        disk            N/A                                              False
capacity_inc    1.00            Processor capacity increment                     False
capped          true            Partition is capped                              False
```

```
conslogin       enable          System Console Login                            False
cpuguard        enable          CPU Guard                                       True
dedicated       true            Partition is dedicated                          False
ent_capacity    4.00            Entitled processor capacity                     False
frequency       93750000        System Bus Frequency                            False
fullcore        false           Enable full CORE dump                           True
fwversion       IBM,SPH01316    Firmware version and revision levels            False
iostat          false           Continuously maintain DISK I/O history          True
keylock         normal          State of system keylock at boot time            False
max_capacity    4.00            Maximum potential processor capacity            False
max_logname     20              Maximum login name length at boot time          True
maxbuf          20              Maximum number of pages in block I/O BUFFER CACHE True
maxmbuf         0               Maximum Kbytes of real memory allowed for MBUFS  True
maxpout         0               HIGH water mark for pending write I/Os per file  True
maxuproc        128             Maximum number of PROCESSES allowed per user     True
min_capacity    1.00            Minimum potential processor capacity            False
minpout         0               LOW water mark for pending write I/Os per file   True
modelname       IBM,7044-270    Machine name                                    False
ncargs          6               ARG/ENV list size in 4K byte blocks             True
pre430core      false           Use pre-430 style CORE dump                     True
pre520tune      disable         Pre-520 tuning compatibility mode               True
realmem         3145728         Amount of usable physical memory in Kbytes      False
rtasversion     1               Open Firmware RTAS version                      False
sec_flags       0               Security Flags                                  True
sed_config      select          Stack Execution Disable (SED) Mode              True
systemid        IBM,0110B5F5F   Hardware system identifier                      False
variable_weight 0               Variable processor capacity weight              False
ngroups_allowed 128             Number of Groups Allowed at boot time           True
$
```

Csoportok megengedett számának lekérése a kernelből: A sys_param szubrutin használatával kérheti le a v_ngroups_allowed paramétert a kernelből.

```
#include<sys/types.h>
#include<sys/var.h>
#include<errno.h>
main()
{
 int rc;
 struct vario myvar;

 rc = sys_parm (SYSP_GET, SYSP_V_NGROUPS_ALLOWED, &myvar);

 if (!rc)
 printf("Number of Groups Allowed = %d\n",
 myvar.v.v_ngroups_allowed.value);
 else
 printf("sys_parm() failed rc = %d, errno = %d\n", rc, errno);
 }
```

Csoportok megengedett számának módosítása az ODM adatbázisban: A Csoportok megengedett száma értéket a kernelben a rendszerbetöltési fázisban kell beállítani. A **chdev** paranccsal módosíthatja az értéket az ODM adatbázisban. Ez a módosítás a rendszer újraindításakor lép érvénybe.

Ha módosítani szeretné a **v_ngroups_allowed** paramétert az ODM adatbázisban a **chdev** paranccsal, akkor írja be a következőt:

```
$ chdev -l sys0 -a ngroups_allowed=2048
sys0 changed
$
```

Szerep alapú hozzáférés-felügyelet

A rendszer adminisztrációja egy fontos szempontja a napi műveleteknek, és a biztonság egy örökös része a legtöbb rendszeradminisztrációs tevékenységnek. Továbbá, az operációs rendszer járulékos védelme érdekében szükséges, hogy a rendszertevékenységeket közel naponta figyelje meg.

A legtöbb rendszer azt igényli, hogy különböző felhasználók kezeljenek különböző rendszeradminisztrációs műszakokat. Szükséges a műszakok szétválasztottságának a fenntartása, hogy egy egyszeri rendszerfelügyelet ne tudja véletlenül vagy végzetesen kikerülni a rendszerbiztonságot. Míg a hagyományos UNIX rendszeradminisztráció nem képes teljesíteni ezeket a célokat, a szerep alapú hozzáférés-felügyelet igen.

Hagyományos UNIX adminisztrációs korlátozás

Az RBAC megold néhány hagyományos UNIX rendszeradminisztrációs problémát. Ezek a problémák a következők:

root adminisztrátori fiók

Hagyományosan az AIX és más UNIX operációs rendszerek rendelkeznek egy **root** nevű rendszeradminisztrátorral (normális esetben a 0 UID jelöli), amely végre tudja hajtani az összes privilegizált rendszeradminisztrációs feladatot a rendszeren. Az, hogy egy felhasználó végzi az összes rendszeradminisztrációs feladatot, a kötelességek elkülönítésével kapcsolatos probléma. Bizonyos környezetekben egy adminisztrátori fiók elfogadott, számos környezet azonban több adminisztrátort igényel, amelyek mindegyike adminisztrátori felelősséggel rendelkezik a különböző rendszeradminisztrációs feladatokhoz.

Az adminisztrátori felesősségek több rendszerfelhasználó közötti megosztásához az eddigi gyakorlat a root felhasználó jelszavának megosztása vagy másik felhasználó létrehozása a root felhasználóéval megegyező UID azonosítóval. A rendszeradminisztrátori kötelességek ezen módszere biztonsági problémákat eredményez, mivel minden adminisztrátor teljes rendszervezérléssel rendelkezik és nincs mód az adminisztrátorok által végrehajtható műveletek korlátozására. Mivel a root felhasználó a legtöbb jogosultsággal rendelkező felhasználó, a root felhasználók jogosulatlan műveleteket is végrehajthatnak és ezen tevékenységek megfigyelését törölhetik, ezáltal az adminisztrátori tevékenységek nem követhetők nyomon.

Jogosultságkiterjesztés SUID segítségével

A UNIX operációs rendszerek hozzáférés-felügyelete eddig a folyamathoz rendelt UID azonosítóval került végrehajtásra a hozzáférés meghatározása érdekében. A 0 root UID azonban hagyományosan kihagyhatta a jogosultságellenőrzéseket. Ezáltal a root felhasználóként futtatott folyamat minden hozzáférés-ellenőrzésen átmehet és minden műveletet végrehajthat. Ez a UNIX **setuid** alkalmazásának biztonsági problémája.

A **setuid** alapelv lehetővé teszi a parancs meghívó felhasználóétól eltérő azonosság alatti futtatását. Erre akkor van szükség, ha egy normál felhasználónak privilegizált feladatot kell végrehajtania. Erre példa az AIX **passwd** parancs. Mivel a normál felhasználó nem fér hozzá a felhasználói jelszavakat tároló fájlhoz, további jogosultság szükséges a felhasználó jelszavának módosításához, így a **passwd** parancs **setuid** a root felhasználóhoz. Ha normál felhasználó futtatja a **passwd** parancsot, akkor az operációs rendszer számára úgy látszik, hogy a root felhasználó fér hozzá a fájlhoz és a hozzáférés biztosított.

Mivel ez az alapelv biztosítja a kívánt funkcionalitást, ezért tulajdonságaiból következően kockázatot jelent. Mivel a **setuid** program valójában root kontextusban fut, ha egy támadó sikeresen átveszi a programot annak kilépése előtt, akkor a támadó rendelkezik a root összes jogosultságával és át tudja lépni az összes operációsrendszer-hozzáférési ellenőrzést és minden műveletet végrehajthat. Jobb megoldás, ha a root felhasználó jogosultságainak csak egy részhalmaza kerül hozzárendelésre a programhoz, így a rendszer a “Legkevesebb jogosultság elv” oldalszám: 80 eljárást követi és a fenyegetés csökken.

Az RBAC elemei

Az RBAC lehetővé teszi szerepek létrehozását a rendszeradminisztrációhoz és az adminisztrációs feladatok kiosztásához a megbízható rendszerfelhasználók halmazán keresztül. AIX rendszeren az RBAC egy mechanizmust biztosít, amelyen keresztül az általában a root felhasználó számára fenntartott adminisztrációs funkciók normál rendszerfelhasználókhoz rendelhetők.

Az RBAC ezt egy szervezeten belül feladatfunkciók (szerepek) meghatározásával és a szerepek felhasználókhoz rendelésével éri el. Az RBAC alapvetően egy keretrendszer, amely a rendszeradminisztrációt szerepek használatán keresztül teszi lehetővé. A szerepek jellemzően a környezet adminisztrációs nézőpontjainak kezeléséhez tartozó

hatókörrel vannak meghatározva. Egy szerep felhasználóhoz rendelése gyakorlatilag engedélyek vagy jogosultságok és hatáskörök halmazát rendeli a felhasználóhoz. Egy kezelési szerep például kezelheti a fájlrendszereket, míg egy másik engedélyezheti felhasználói fiókok létrehozását.

Az RBAC adminisztráció a következő előnyökkel rendelkezik a hagyományos UNIX adminisztrációval összehasonlítva:

- A rendszeradminisztrációt több felhasználó végezheti a fiókhozzáférés megosztása nélkül.
- Biztonsági elkülönítés részletes adminisztráción keresztül, mivel a szükségesnél egyik adminisztrátornak sem kell több jogosultságot kapnia.
- Engedélyezi a legkevesebb jogosultság biztonsági modell betartatását. A felhasználók és alkalmazások csak akkor kapják meg a szükséges jogosultságokat, amikor szükséges, így csökken a rendszer támadója által okozható kár.
- Engedélyezi vállalati szintű biztonsági irányelvek megvalósítását és következetes betartatását a rendszerfelügyelet és hozzáférés-felügyelet tekintetében.
- A szerepmeghatározás egyszeri létrehozás után felhasználókhoz rendelhető vagy elvehető azoktól, feladataik változásának megfelelően.

Az RBAC keretrendszer a következő három fogalmat állítja a középpontba:

- Felhatalmazások
- Szerepek
- Jogosultságok

Ez a három fogalom együtt lehetővé teszi az RBAC rendszernek a legkevesebb jogosultság elvének betartatását.

Felhatalmazások:

A felhatalmazás biztonsággal kapcsolatos funkciókhoz vagy parancsokhoz tartozó karaktersorozat. A felhatalmazások lehetőséget adnak jogok megadására felhasználóknak privilegizált műveletek végrehajtására és különböző felhasználóosztályoknak különböző szintű funkcionalitás biztosításához.

Felhatalmazás által irányított parancs futtatásakor a hozzáférés csak akkor kerül megadásra, ha a felhasználó rendelkezik a szükséges felhatalmazással. A felhatalmazás egy kulcsként képzelhető el, amely hozzáférést ad néhány parancshoz. A hitelesítések nem közvetlenül a felhasználóhoz kerülnek társításra. A felhasználókhoz szerepek rendelhetők, amelyek felhatalmazások gyűjteményei.

Szerepek:

A szerepek lehetővé teszik a rendszerben lévő kezelési funkciók egy halmazának csoportosítását. Annak a hasonlatnak a használatával, amelyben a felhatalmazás egy kulcs, a szerep kulcscsomónak tekinthető, amely több felhatalmazást tartalmazhat. A felhatalmazások közvetlenül hozzárendelhetők egy szerephez, vagy közvetve egy alszerepen keresztül. Az alszerep egyszerűen egy másik szerep, amelytől egy adott szerep a felhatalmazásokat örökli.

A szerep maga nem biztosít a felhasználó számára további jogokat. Ehelyett gyűjtési mechanizmusként szolgál a felhatalmazásokhoz és szolgáltatásként felhatalmazások felhasználóhoz rendeléséhez. Szerep megadása és a szerep felhasználóhoz rendelése meghatározza a felhasználó által végrehajtható rendszeradminisztrációs feladatokat. Szerep hozzárendelése után a szerepadminisztrátor hozzá tudja rendelni a szerepet egy vagy több felhasználóhoz a szerep által képviselt privilegizált műveletek kezelése érdekében. Egy felhasználóhoz több szerep rendelhető. A szerep felhasználóhoz rendelése után a felhasználó a szerephez tartozó felhatalmazások segítségével feloldhatja a rendszeren lévő adminisztrációs parancsok elérésének zárolását.

A szervezeti irányelvek és eljárások szerepek felhasználóhoz rendelésének módját határozzák meg. Ne rendeljen túl sok felhatalmazást egy szerephez, és ne rendeljen egy szerepet túl sok felhasználóhoz. A legtöbb szerep csak az adminisztrátori csapat tagjaihoz rendelhető. Ahhoz hasonlóan, hogy a root jogosultságait csak megbízható felhasználók kapták, szerepek is csak megbízható felhasználókhoz rendelhetők. Szerepeket csak az indokolható igényekkel

rendelkező felhasználóknak adjon, és csak a szükséges időtartamra. Ez a gyakorlat csökkenti annak esélyét, hogy jogosulatlan felhasználó felhatalmazásokat szerezzen vagy foganatosítson.

Jogosultságok:

A jogosultság egy folyamatattribútum, amely lehetővé teszi, hogy a folyamat átlépjen adott rendszermegszorításokat és -korlátozásokat.

A feljogosítási mechanizmus a megbízható alkalmazások számára a nem megbízható alkalmazások számára nem megengedett képességeket biztosít. A jogosultságok segítségével például felülbírálhatók a biztonsági megszorítások, adott rendszererőforrások, mint például a memória- és lemezterület, kiterjesztett használatának lehetővé tétele érdekében, és beállítható a folyamat teljesítménye és prioritása. A jogosultság olyan képességként képzelhető el, amely lehetővé teszi, hogy a folyamat legyőzze a rendszer egy adott megszorítását.

A felhatalmazások és szerepek felhasználószintű eszközök, amelyek biztosítják egy felhasználó számára a privilegizált műveletek elérését. Más részről a jogosultságok a kernelben használt megszorítási mechanizmusok annak meghatározásához, hogy egy folyamat végrehajthat-e egy adott tevékenységet.

A folyamathoz jogosultságok vannak rendelve, amelyek jellemzően privilegizált parancs meghívásán keresztül biztosítottak. Ezekkel a hozzárendelt jogosultságokkal a folyamat végre tudja hajtani a kapcsoló privilegizált műveletet. Ha például egy felhasználó olyan szerepet használ, amely jogosult egy parancs futtatására, akkor a parancs futtatásakor jogosultságok halmaza kerül hozzárendelésre a folyamathoz.

Legkevesebb jogosultság elv:

Az operációs rendszerekben egyes műveletek privilegizáltak és ezen műveletek végrehajtása a felhatalmazott felhasználókra van korlátozva. A privilegizált műveletek általában olyan feladatokat tartalmaznak, mint például a rendszer újraindítása, fájlrendszerek hozzáadása és módosítása, felhasználók hozzáadása és módosítása, valamint a rendszer dátumának és idejének módosítása.

A hagyományos UNIX rendszerekben egy folyamat illetve felhasználó normál vagy privilegizált (superuser vagy root) módban lehet. A root nevében futó folyamat tetszőleges parancsot és rendszerműveletet végrehajthat, míg a normál felhasználó nem hajthatja végre a privilegizált műveleteket. A hagyományos UNIX rendszer nagyon durva 'mindent vagy semmit' jogosultság-alapelvvel rendelkezik és a túlzott jogosultságokkal rendelkező adminisztrátor biztonsági fenyegetésével kell szembenéznie.

A hagyományos UNIX megközelítés, amelyben egyetlen privilegizált mód minden hozzáférést megad a rendszerhez, túlságosan durva a nagyon biztonságos rendszerek szükségleteinek kielégítéséhez. A biztonságosnak tervezett rendszer megköveteli, hogy minden folyamat a feladat végrehajtásához szükséges jogosultságok legjobban korlátozott halmazát kapja. A jogosultságok használata azzal az előnnyel jár, hogy csak a bizonyos jogosultságokat igénylő folyamatok számára kell biztosítani ezeket a jogosultságokat. A jogosultságok ezen korlátozása legkevesebb jogosultság elveként ismert és hasznos az óvatlan vagy rosszindulató adminisztrátorok valamint operátorok által okozott rendszerkárosodás korlátozásához.

A jelszó módosítása például adott jogosultságokat igényel a normál felhasználó számára el nem érhető fájlok eléréséhez. Ha a felhasználók mindig rendelkeznének ezekkel a jogosultságjogosultságokkal, akkor más, biztonsági szempontból nem kívánatos műveleteket is végrehajthatnának. Emiatt a szükséges jogosultságokat csak a **passwd** parancs kapja meg és nem az összes felhasználó.

RBAC környezetben maguk a felhasználók nem rendelkeznek örökölt jogosultságokkal. A felhasználók egyszerűen futtathatnak bizonyos parancsokat, amelyek aztán megkapják a jogosultságokat. Ha a felhasználó ehelyett közvetlenül megkapta volna a jogosultságokat, azokat bármikor, tetszőleges célra felhasználhatta volna. A jogosultságok egyedi parancsokra korlátozása lehetővé teszi azon kontextus korlátozását, amelyben a jogosultságok érvényesek. Ez a biztonság növekedéséhez vezet, ha ugyanis egy megbízható alkalmazás hibáját kihasználja egy támadó, akkor a támadó csak jogosultságok korlátozott halmazával fog rendelkezni a root teljes hatásköre és összes jogosultsága helyett.

A megbízható alkalmazásokat alaposan meg kell vizsgálni jogosultságok biztosítása előtt. Ezen kívül a jogosultságokat csak akkor és ott kell megadni, ahol az alkalmazás igényli azt. A megbízható alkalmazások hasonlók a többi programhoz, az egyetlen különbség, hogy a megbízható alkalmazások végrehajthatnak olyan műveleteket, amelyeket a megbízhatatlan alkalmazások nem.

AIX RBAC

Az AIX korlátozott RBAC megvalósítást biztosított az AIX 6.1 változat előtt.

Az AIX 6.1 változattól kezdve az RBAC új megvalósítása nagyon részletes mechanizmust biztosít a rendszeradminisztrációs feladatok felosztására. Mivel ez a két RBAC megvalósítás jelentősen eltér funkcionalitásban, a következő kifejezések kerülnek felhasználásra:

Örökölt RBAC mód
: Az AIX szerepek történelmi viselkedése, amely az AIX 6.1 előtti változatokra vonatkozik

Kiterjesztett RBAC mód
: Az AIX 6.1 változatban bevezetett új megvalósítás

Mindkét működési mód támogatott. Az újonnan telepített AIX 6.1 rendszereken azonban a kiterjesztett RBAC mód az alapértelmezett. A következő szakaszok röviden leírják a két módot és azok eltéréseit, valamint információkat adnak a rendszer konfigurálásához a kívánt módban történő működtetéshez.

Örökölt RBAC mód:

Az AIX 6.1 változat előtt az AIX korlátozott RBAC funkcionalitást biztosított, ami lehetővé tette a nem root felhasználók számára bizonyos rendszeradminisztrációs feladatok végrehajtását.

Ebben az RBAC megvalósításban adott adminisztrációs parancs nem root felhasználó által történő meghívásakor a parancs kódja meghatározza, hogy a felhasználó rendelkezik-e a szükséges felhatalmazású szereppel. Találat esetén a parancs végrehajtása folytatódik. Ellenkező esetben a parancs hibával meghiúsul. Gyakran szükséges, hogy egy parancsot felügyelő felhatalmazás **setuid** legyen a root felhasználóra nézve, hogy a felhatalmazott hívó rendelkezzen a művelet végrehajtásához szükséges jogosultságokkal.

Ez az RBAC megvalósítás bevezette felhatalmazások előre meghatározott, de a felhasználók által bővíthető halmazát, amelyek segítségével meghatározható a hozzáférés az adminisztrációs parancsokhoz. Ezen kívül adminisztrációs parancsok és felületek szerepek létrehozására, felhatalmazások szerepekhez rendeléséhez és szerepek felhasználókhoz rendelésére szolgáló keretrendszere is biztosított.

Míg ez a megvalósítás lehetőséget ad a rendszeradminisztrátori felelősségek megosztására, a következő megszorítások mellett működik:

1. A keretrendszer a parancsok és alkalmazások módosítását igényli, hogy azok képesek legyenek az RBAC használatára.
2. Az előre meghatározott felhatalmazások nem elég részletesek és a felhatalmazások létrehozására szolgáló mechanizmusok nem robusztusak.
3. Parancs futtatásához gyakran szükséges a tagság bizonyos csoportban, valamint egy adott felhatalmazású szerep.
4. A feladatok elkülönítését nehéz megvalósítani. Ha egy felhasználó több szerepet is kap, nincs lehetősége egyetlen szerep birtokosaként cselekedni. A felhasználó mindig rendelkezni fog az összes szerepének összes felhatalmazásával.
5. A legkevesebb felhatalmazás elve nincs elfogadva az operációs rendszerben. A parancsoknak jellemzően rendelkezniük kell SUID bittel a root felhasználóhoz.

Az örökölt RBAC mód a kompatibilitás miatt támogatott, de az alapértelmezett RBAC mód a kiterjesztett RBAC mód. Az AIX rendszereken a kiterjesztett RBAC mód az előnyben részesített.

Kiterjesztett RBAC mód:

Az AIX 6.1 az RBAC hatékonyabb megvalósítását biztosítja. A bizonyos műveletekhez adminisztrátori jogosultságokat igénylő alkalmazások új integrációs lehetőségeket kaptak a kiterjesztett AIX RBAC infrastruktúrával.

Ezek az integrációs lehetőségek a finom jogosultságok és felhatalmazások használatát állítják középpontba, valamint lehetőséget ad a rendszer bármely parancsának beállítására privilegizált parancsként. A kiterjesztett RBAC mód szolgáltatásai az AIX minden alapértelmezett telepítésén telepítésre és engedélyezésre kerülnek az AIX 6.1 változattól kezdődően.

A kiterjesztett RBAC mód felhatalmazások, szerepek, jogosultságok, privilegizált parancsok, eszközök és fájlok konfigurálható halmazát biztosítja az alább felsorolt RBAC adatbázisokban. A kiterjesztett RBAC használatával az adatbázisok elhelyezkedhetnek a helyi fájlrendszeren vagy felügyelhetők távolról LDAP- keresztül.

- Felhatalmazási adatbázis
- Szerepadatbázis
- Privilegizált parancsadatbázis
- Privilegizált eszközadatbázis
- Privilegizált fájladatbázis

A kiterjesztett RBAC mód a felhatalmazások új elnevezési megállapodását vezeti be, amely lehetővé teszi a felhatalmazások hierarchiájának létrehozását. Az AIX rendszer által meghatározott felhatalmazások részletes halmazát biztosítja, valamint az adminisztrátor szükség esetén szabadon létrehozhat további felhasználó által megadott felhatalmazásokat.

A szerepek viselkedése továbbfejlesztésre került a feladatfunkcionalitás elkülönítésének biztosításához. A kiterjesztett RBAC bevezeti a szerepmunkamenetek fogalmát. A szerepmunkamenet legalább egy társított szereppel rendelkező folyamat. A felhasználó bármely hozzárendelt szerephez létrehozhat szerepmunkamenetet, így aktiválva egyszerre egy vagy több kijelölt szerepet. Alapértelmezésben az új folyamatokhoz nincs szerep társítva. A szerepek kiterjesztésre kerültek azon követelmény támogatása érdekében, hogy a felhasználónak hitelesítenie kell magát a szerep aktiválása előtt, így védekezhet a felhasználói munkamenetet átvevő támadó ellen, akinek hitelesítenie kellene magát a felhasználó szerepeinek aktiválásához.

A privilegizált parancsadatbázis bevezetése megvalósítja a legkevesebb jogosultság elvét. A rendszerjogosultságok részletessége növekedett és a jogosultságok kifejezetten egy parancsnak adhatók, valamint a parancs végrehajtása felhatalmazással felügyelhető. Ez a parancs kódjának módosítása nélkül biztosítja a parancsvégrehajtáshoz a felhatalmazásellenőrzéseket kikényszerítő funkcionalitást. A privilegizált parancsadatbázis használata megszünteti a SUID és SGID alkalmazások szükségességét, mivel lehetőség van csak a szükséges jogosultságok hozzárendelésére.

A privilegizált eszközadatbázis lehetővé teszi az eszközhozzáférés jogosultságalapú irányítását, míg a privilegizált fájladatbázis a felhatalmazások alapján lehetővé teszi a jogosulatlan felhasználók fájlhozzáférésének korlátozását. Ezek az adatbázisok növelik az egyébként felhatalmazással nem rendelkező felhasználókhoz rendelhető rendszeradminisztrációs feladatok finomságát.

Az RBAC adatbázisokban található információk összegyűjtés és ellenőrzés után a kernel által kernel biztonsági táblázatok (KST) néven kijelölt területre kerülnek elküldésre. Fontos megjegyezni, hogy a KST által tartalmazott adatok állapota meghatározza a rendszer biztonsági irányelveit. A felhasználói szintű RBAC adatbázisokban módosított bejegyzések nem kerülnek felhasználásra biztonsági döntésekhez, amíg ezek az információk a **setkst** parancs használatával nem kerülnek elküldésre a KST-be.

Az RBAC mód beállítása:

Az RBAC módot a kernel rendszerszintű konfigurációs változója vezérli. Ez a változó megadja, hogy a kiterjesztett RBAC mód engedélyezve van-e.

A kiterjesztett RBAC mód AIX 6.1 és újabb rendszereken alapértelmezésben be van kapcsolva . A **chdev** parancs futtatásával a **sys0** eszközön és az **enhanced_RBAC** attribútum **false** értékre állításával letilthatja a kiterjesztett RBAC módot és visszatérhet az örökölt RBAC módba. Az **enhanced_RBAC** attribútum módosításának életbe léptetéséhez újra kell indítania a rendszert. A kiterjesztett RBAC mód engedélyezéséhez az **enhanced_RBAC** attribútumot **true** értékre kell állítani. A mód programozói eszközökkel is beállítható vagy lekérdezhető a **sys_parm()** rendszerhíváson keresztül.

Futtassa a következő parancsot a rendszeren az aktuális RBAC mód lekéréséhez:

```
lsattr -E -l sys0 -a enhanced_RBAC
```

A következő parancs futtatásával, majd a rendszer újraindításával letilthatja a kiterjesztett RBAC módot:

```
chdev -l sys0 -a enhanced_RBAC=false
```

WPAR környezetben az RBAC mód csak a globális rendszerből állítható be és a globális módot valamint a rendszeren található összes WPAR-t is érinti.

Az örökölt RBAC mód és a kiterjesztett RBAC mód összehasonlítása:

A meglévő és új felületek módosításra kerültek a rendszerkonfiguráció ellenőrzése és az új kód futtatása vagy a régi viselkedés követése érdekében.

Örökölt RBAC módban csak magában a parancs kódjában ellenőrzött felhatalmazások kerülnek kikényszerítésre. A kernel biztonsági táblázatai (KST) nincsenek hatással a parancs végrehajtására vagy a felhatalmazásellenőrzésekre. Annak meghatározása, hogy a felhasználó rendelkezik-e felhatalmazással, az örökölt RBAC módú viselkedést követi. A felhasználó felhatalmazásai lekérésre kerülnek és a rendszer egyezést keres. Az új szolgáltatások, mint például az **swrole** parancs, és a **default_roles**, illetve **auth_mode** attribútum nem érhető el örökölt RBAC módban. Azonban az új jogosultságok, felhatalmazások és felügyeleti parancsok örökölt RBAC módban támogatottak.

Az alábbi táblázat felsorolja az örökölt és a kiterjesztett RBAC mód közötti különbségek egy részét.

9. táblázat: Különbségek az örökölt és kiterjesztett RBAC módok között

Szolgáltatás	Örökölt RBAC	Kiterjesztett RBAC
Szerepaktiválás	Mindig aktív a felhasználó minden szerepe	Alapértelmezésben a szerepek nem aktívak az **swrole** paranccsal történő kifejezett elfogadásukig.
default_roles attribútum	Nem érhető el	Támogatott
swrole parancs	Nem érhető el	Támogatott
Szerepkezelő parancsok	Támogatott	Támogatott
Felhatalmazáskezelő parancsok	Támogatott	Támogatott
Felhatalmazási hierarchia	Minden felhatalmazás független. Nincs hierarchiafunkcionalitás.	Támogatja a felhatalmazáshierarchia fogalmát, ahol a felhatalmazások más felhatalmazások szülői lehetnek
Felhatalmazásellenőrzések	Csak akkor kerül betartatásra, ha a parancs maga ellenőrzi a felhatalmazást	A privilegizált parancsadatbázison és/vagy magán a parancson keresztül kerül betartatásra.
Finom jogosultságok	Támogatott	Támogatott
pvi parancs	Nem érhető el	Támogatott
Kernel biztonsági táblázatok	Nem érhető el	Támogatott
RBAC adatbázis helye	Helyi fájlok	Helyi fájlok vagy LDAP

Kiterjesztett RBAC használata

A rendszeradminisztrátoroknak a kiterjesztett RBAC használatához a következő területeket kell ismerniük.

RBAC felhatalmazások:

A felhatalmazások a szerep alapú hozzáférés-felügyelet (RBAC) fontos részét képezik. Az operációs rendszer felhatalmazási karaktersorozatokat használ az alkalmasság meghatározására privilegizált művelet végrehajtása előtt. A kapcsolódó ellenőrzések végrehajthatók a kódból explicit módon vagy végrehajthatja a betöltő a védett privilegizált végrehajtható fájlok futtatásakor.

A felhatalmazási karaktersorozatok elnevezése jelzi az általuk ábrázolt és vezérelt privilegizált műveletet. Az AIX felhatalmazáselnevezési megállapodása támogatja a hierarchikus szerkezetet, amelyet a felhatalmazás szöveges neve jelöl. Az AIX felhatalmazási karaktersorozat pontozott jelölési formát használ a felhatalmazási hierarchia leírásához. Új fájlrendszer létrehozására szolgáló felhatalmazás például az **aix.fs.manage.create**. Ha ezt a felhatalmazást tartalmazza egy szerep, akkor a szerepet hozzárendelő felhasználó létre tudja hozni az AIX fájlrendszereket. Ha az **aix.fs.manage** szülő felhatalmazást tartalmazza egy szerep, akkor a szereppel rendelkező felhasználó más fájlrendszer-kezelési feladatot is végrehajthat, és fájlrendszereket is létre tud hozni.

Az AIX RBAC megkülönbözteti a rendszer által biztosított (rendszer által meghatározott) és a felhasználó által telepítés után létrehozott (felhasználó által megadott) felhatalmazásokat.

Rendszer által meghatározott felhatalmazások:

Az AIX előre meghatározott és nem módosítható felhatalmazásokat biztosít. Ezek rendszer által meghatározott felhatalmazásokként ismertek. Ezek a felhatalmazások különböző privilegizált AIX műveletekhez vannak rendelve; a hozzárendelés a privilegizált általános adatbázisban van megadva.

A rendszer által meghatározott felhatalmazási hierarchia tetején az **aix** felhatalmazás található. Ez a felhatalmazás az összes többi rendszer által meghatározott felhatalmazás szülője. Felhatalmazás szerephez adása minden rendszer által meghatározott felhatalmazást a szerephez ad. A teljes AIX rendszer által meghatározott felhatalmazáshalmaz és az egyes felhatalmazások rövid leírásának megjelenítéséhez futtassa a következő parancsot:

```
lsauth –f –a description ALL_SYS
```

A fenti parancs kimenete megjeleníti, hogy a rendszer által meghatározott felhatalmazások listája többszintű hierarchia. Az **aix** felhatalmazás például számos közvetlen leszármazottal rendelkezik. Ezen leszármazottak mindegyike másik hierarchia szülője. Az **aix.fs** felhatalmazás több leszármazott felhatalmazást tartalmaz, az **aix.fs.manage** felhatalmazást is beleértve, amely több felhatalmazást tartalmaz, mint például az **aix.fs.manage.change** és **aix.fs.manage.create**.

Felhasználó által megadott felhatalmazások:

A rendszer által meghatározott felhatalmazásokon felül az AIX RBAC lehetővé teszi, hogy a rendszeradminisztrátorok saját egyéni felhatalmazásokat adjanak meg a felhatalmazási adatbázisban (/etc/security/authorizations). Ezeket felhasználó által megadott felhatalmazásoknak hívják.

A rendszeradminisztrátor felhasználó által megadott felhatalmazásokat vehet fel, módosíthat és törölhet. A rendszeradminisztrátor például lehetővé teheti néhány felhasználó számára privilegizált parancs futtatását felhasználó által megadott felhatalmazás létrehozásával, majd a felhatalmazás parancshoz rendelésével és felhasználókhoz rendelt szerephez adásával.

A felhasználó által megadott felhatalmazások ugyanazt a hierarchiát támogatják, mint a rendszer által meghatározott felhatalmazások. Az AIX felhasználó által megadott felhatalmazások elnevezésére korlátozások érvényesek.

- A felhasználó által megadott felhatalmazásokat egy új felső szintű szülő alatt kell megadni. Más szavakkal a felhasználó által megadott felhatalmazások nem lehetnek rendszer által meghatározott felhatalmazások leszármazottai (**aix**).
- A felhatalmazásnév maximum 63 nyomtatható karaktert tartalmazhat.
- A felhatalmazás szülő hierarchiája maximum nyolc szintet tartalmazhat.
- A felhatalmazás tetszőleges számú közvetlen leszármazottal rendelkezhet, de csak egy közvetlen szülővel. Két független felhatalmazás nem rendelkezhet azonos közvetlen leszármazottal.

Mivel a hierarchia nem engedi, hogy az elem több közvetlen szülővel rendelkezzen, nem hozható létre felhasználó által megadott felhatalmazás, amely a meglévő rendszer által meghatározott felhatalmazás szülője. Az **aix.custom** nevű felhatalmazás létrehozására történt kísérlet meghiúsul és az **custom.aix** nevű felhatalmazás létrehozása teljesen új felhatalmazást eredményez, valamint nem az **aix** rendszer által meghatározott felhatalmazás szülőjeként funkcionál.

A következő szintaxis javasolt a felhasználó által megadott felhatalmazások létrehozásakor több szoftverösszetevőn a felhatalmazásnevek közötti ütközés elkerülése érdekében:

szállítónév.*terméknév*.*funkció*.*funkció1*.*funkció2*...

szállítónév
: A szoftvermodul szállítójának nevét azonosítja.

terméknév
: Az RBAC által kezelt termék magas szintű termékneve.

funkció, *funkció1*, *funkció2* **...**
: Ezek a karaktersorozatok az RBAC által kezelendő funkciókat ábrázolják. Ezek a karaktersorozatok a funkciók szervezésének hierarchikus ábrázolását is biztosítja.

Az **ibm.db2.manage** például az IBM DB2 adatbáziscsomag kezelési szempontjait ábrázolhatja. Ahogy korábban említettük, az **aix** *szállítónév* karaktersorozat AIX általi használatra van fenntartva és felhasználó által megadott felhatalmazásokhoz nem megengedett.

A rendszeradminisztrátorok számos felhatalmazás-kezelési parancsot használhatnak a felhasználó által megadott felhatalmazások listázásához, létrehozásához, módosításához és eltávolításához. A felhasználó által megadott felhatalmazások az **mkauth** paranccsal létrehozhatók, a **chauth** paranccsal módosíthatók, az **rmauth** paranccsal eltávolíthatók és az **lsauth** paranccsal megjeleníthetők. Az összes felhasználó által megadott rendszerfelhatalmazás és rövid leírásának megjelenítéséhez futtassa a következő parancsot:

```
lsauth –f –a description ALL_USR
```

Felhasználó által megadott felhatalmazás létrehozása előtt vegye figyelembe a következő problémákat:

- Új felhasználó által megadott felhatalmazás helyett meglévő rendszer által meghatározott felhatalmazás használata megfelelő-e?
- Az új felhatalmazás meglévő felhasználó által megadott felhatalmazáshierarchia alatt található, vagy egy új hierarchia első felhatalmazása?
- Ha ez új hierarchia, akkor mi a szerkezete?
- Mi a felhatalmazás szöveges leírása?
- Szükség van a felhatalmazás leírásának fordítására?
- A felhatalmazás létrehozásakor van ok adott felhatalmazásazonosító megadására? A felhatalmazásazonosító előállításához ajánlatos az **mkauth** parancsot használni.

A problémák átgondolása után tegye a következőket a felhatalmazás létrehozásához:

1. Ha nyelvfordítás szükséges, akkor hozza létre vagy adja hozzá a leírást az üzenetkatalógushoz.
2. Az **mkauth** parancs segítségével hozza létre az összes szülőfelhatalmazást a hierarchiában, ha még nem léteznek.
3. Az **mkauth** parancs segítségével hozza létre a kívánt felhatalmazást. Adja meg az **id** attribútumot a paranccsal, ha egy adott érték szükséges.

Örökölt felhatalmazás átállítása:

Az AIX Version 6.1 előtt az operációs rendszer felhatalmazások korlátozott, előre meghatározott halmazával rendelkezett. Ezek a felhatalmazások nem a rendszer valamely fájljában voltak meghatározva, de szerepekhez lehetett rendelni őket. Ezen örökölt felhatalmazások támogatása érdekében az új AIX Version 6.1 és újabb RBAC keretrendszerben az örökölt felhatalmazások felhasználó által megadott felhatalmazásokként vannak meghatározva és a felhatalmazási adatbázis alapértelmezésben tartalmazza őket.

Mivel az AIX operációs rendszer új felhatalmazás elnevezési megállapodás felé mozdul el, az AIX operációs rendszer régi felhatalmazási nevei módosításra kerültek a vonatkozó új felhatalmazás további ellenőrzésével és és a folyamathoz valamely felhatalmazás megléte esetén a hozzáférés biztosítása érdekében. A következő táblázat felsorolja az örökölt, előre meghatározott felhatalmazásokat és a megfelelő, új rendszer által meghatározott felhatalmazásokat.

Meglévő AIX felhatalmazás	Megfelelő új felhatalmazás
Backup	aix.fs.manage.backup
Diagnostics	aix.system.config.diag
DiskQuotaAdmin	aix.fs.manage.quota
GroupAdmin	aix.security.group
ListAuditClasses	aix.security.audit.list
PasswdAdmin	aix.security.passwd
PasswdManage	aix.security.passwd.normal
UserAdmin	aix.security.user
UserAudit	aix.security.user.change
RoleAdmin	aix.security.role
Restore	aix.fs.manage.restore

RBAC szerepek:

A szerep felhatalmazások felhasználóhoz rendelésére és rendszeradminisztrátori feladatok csoportosítására szolgál mechanizmus. Az AIX szerepek elsődlegesen a felhatalmazások gyűjteményének tárolói.

Az AIX támogatja a felhatalmazások közvetlen hozzárendelését a szerephez vagy közvetett hozzárendelését alszerepeken keresztül. Az alszerep a szerephez a szerep **rolelist** attribútumában adható meg. A szerep beállítása, hogy rendelkezzen egy kijelölt alszereppel, ténylegesen hozzárendeli az alszerep összes felhatalmazását a szerephez.

Szerep felhasználóhoz rendelése lehetővé teszi, hogy a felhasználó hozzáférjen a szerephez és használja a szerep által tartalmazott felhatalmazásokat. A rendszeradminisztrátor egy szerepet több felhasználóhoz is hozzárendelhet és több szerepet is rendelhet egy felhasználóhoz. A több hozzárendelt szereppel rendelkező felhasználó szükség esetén több szerepet tud aktiválni (maximum nyolcat) egyszerre a rendszerkezelési funkciók végrehajtásához.

Az AIX előre meghatározott szerepeket biztosít a rendszerkezeléshez. Várhatón az ügyfeleknek saját egyéni szerepeket létrehozniuk és módosítaniuk kell a meglévő előre meghatározott szerepeket. Számos szerepkezelési parancs áll rendelkezésre az AIX szerepek listázásához, létrehozásához, módosításához és eltávolításához. A szerepek az **mkrole** paranccsal hozhatók létre, a **chrole** paranccsal módosíthatók, az **rmrole** paranccsal távolíthatók el és az **lsrole** paranccsal jeleníthetők meg.

Új AIX szerep létrehozsakor vegye figyelembe a következő problémákat:

- Mi lesz a szerep neve?
- A szerepnév egy szöveges karaktersorozat, amelynek némi betekintést kell nyújtania a szerep képességeibe. A szerepnevek maximum 63 nyomtatható karaktert tartalmazhatnak.
- Milyen felhatalmazások szükségesek a szerephez? Fontolja meg, hogy a felhatalmazásokat közvetlenül kell-e a szerepekhez rendelni, vagy közvetett módon, alszerepeken keresztül.
- A felhasználónak hitelesítenie kell-e magát a szerep aktiválásakor?

Szerep aktiválása:

Alapértelmezésben a kiterjesztett RBAC-t használó AIX Version 6.1 és újabb rendszeren amikor a felhasználó hitelesíti magát a rendszerhez, akkor a felhasználó munkamenetéhez nincsenek szerepek vagy jogosultságok társítva. Szerepek munkamenethez társításához a felhasználónak egy külön hitelesítési parancsot (az **swrole** parancsot) kell meghívnia a szerepre vagy szerepekre váltáshoz.

A felhasználó csak olyan szerepeket aktiválhat, amelyeket korábban hozzárendeltek. Alapértelmezésben a felhasználónak hitelesítenie kell magát szerepmunkamenetbe lépéskor vagy szerep munkamenetéhez adásakor. A szerepek az **auth_mode** szerepattribútummal kijelölhetők, hogy ne igényeljenek hitelesítést.

A váltás egy új szerepmunkamenetre új parancsértelmezőt (munkamenetet) hoz létre a korábbi munkamenet szerepeinek öröklése nélkül. Ez az új szerephez új folyamat-parancsértelmező létrehozásával és az új szerepazonosító (RID) folyamathoz rendelésével történik. Az új munkamenet létrehozása hasonló a **su** parancs használatához, kivéve hogy ebben az esetben csak a folyamat szerepazonosítója változik és az olyan jellemzők, mint az UID vagy GID nem. Az **swrole** parancs lehetővé teszi a felhasználónak egy vagy több szerepből álló szerepmunkamenet létrehozását. Nincs a felhasználót az aktuális szerepmunkamenetről új szerepmunkamenetre váltásban megakadályozó korlátozás. Mivel az új munkamenet egy új folyamat, az új munkamenet nem örököl szerepeket a korábbi munkamenetből. A korábbi munkamenet visszaállítása érdekében a felhasználónak ki kell lépnie az aktuális szerepmunkamenetből. A munkamenetben elfogadott szerepek (az aktív szerephalmaz) a munkamenetben kiadott **rolelist** parancs segítségével sorolhatók fel. Az adminisztrátor is használhatja a **rolelist** parancsot egy adott rendszerfolyamat aktív szerephalmazának felsorolására.

A felhasználóhoz az új **default_roles** felhasználói attribútum segítségével szerepek alapértelmezett halmaza rendelhető. Ezt az attribútumot olyan helyzetekre tervezték, ahol egy adott felhasználó nevében létrehozott folyamatokhoz mindig szerepek adott halmazát kell társítani, például a **cron** parancs esetén. A cron szolgáltatás a háttérben fut és a megadott felhasználó nevében parancsokat futtat. Lehetséges, hogy néhány futtatott parancs felhatalmazást igényel. Ez megköveteli szerepek halmazának állandóan aktívként jelölésének lehetőségét adott felhasználói azonosítóhoz, mivel nincs a **cron** parancshoz megfelelő mechanizmus a szerepek későbbi megszerzésére. A **default_roles** attribútum beállítható akár nyolc szerepnév vagy az **ALL** speciális érték tartalmazására. A **default_roles=ALL** beállítása a felhasználó összes szerepét a munkamenethez rendeli. Ha a felhasználó nyolcnál több szereppel rendelkezik, akkor csak az első nyolc szerep kerül engedélyezésre a munkamenethez.

Szerepek munkamenetenkénti maximális száma:

Kiterjesztett RBAC esetén a rendszeradminisztrátor rendszerszinten beállíthatja az adott szerepmunkamenetben a felhasználó által aktiválható szerepek maximális számát. Alapértelmezésben a felhasználó egy munkamenetben legfeljebb nyolc szerepet aktiválhat.

Bizonyos környezetek a feladatok jobb elkülönítését igényelhetik, amelyben a felhasználó egyszerre csak egyetlen szerepet aktiválhat. Ezekben a környezetekben az /etc/security/login.cfg fájl **usw** szakaszában lévő **maxroles** attribútum módosításával korlátozható a munkamenetenként engedélyezett szerepek száma. A **maxroles** attribútum 1 és 8 közötti értékre állításával megadható a munkamenetenként maximálisan megengedhető szerepek száma.

A munkamenetenkénti szerepek számára vonatkozó korlátozás aktuális értéke a következő parancs futtatásával jeleníthető meg:

```
lssec –f /etc/security/login.cfg –s usw –a maxroles
```

A rendszer módosításához, hogy egy felhasználó egyszerre csak egye szerepet aktiválhasson, futtassa a következő parancsot:

```
chsec –f /etc/security/login.cfg –s usw –a maxroles=1
```

A **maxroles** attribútum értékének módosítása azonnal életbe lép a létrehozott új szerepmunkamenetekben és nem igényli a rendszer újraindítását. Az érték módosítása előtt meglévő szerepmunkameneteket nem érinti a módosítás. A munkamenetenkénti szerepek maximális számának kikényszerítése a munkamenet kezdeményezésekor történik meg.

Előre meghatározott szerepek:

Az Előre meghatározott szerepek halmaza az új és későbbi AIX Version 6.1 telepítés helyi szerepadatbázisában (/etc/security/roles) van meghatározva. A szerepek ezen halmazát a jellemző adminisztrációs felelősségek csoportosítására tervezték.

A szerepek halmaza az adminisztrációs feladatok javasolt felosztásaként szolgál. A szerepadminisztrátorok módosíthatják vagy eltávolíthatják ezeket a szerepeket, illetve új szerepeket hozhatnak létre a környezetük szükségletei szerint. A következő táblázat felsorolja a biztosított szerepeket és az egyes szerepek képességeit.

Szerepnév	**Szerepleírás**
auditadm	Megfigyelési adminisztrátor. Az auditadm szerep a rendszer megfigyelési és naplózási irányelveinek konfigurálásáért felel, a rendszerszintű, felhasználószintű és szerepszintű attribútumokat is beleértve. A szerep hozzáférhet a megfigyelési nyomkövetéshez.
fsadm	Fájlrendszer adminisztrátor. Az fsadm szerep fájlrendszereket hoz létre, és elérhetővé teszi azokat a rendszeren a felhasználók számára. Az fsadm szerep többek között a következőkért felel: • Felépítési irányelvek meghatározása • Megosztási irányelvek • Kvóták hozzárendelése • Tömörítési szint meghatározása • Fájlrendszer formátumok kialakítása • Biztonsági mentési és visszaállítási tevékenységek
isso	Információs rendszer adatvédelmi megbízottja. Az ISSO a szerepek létrehozásáért és társításáért felelős, így a rendszer legerősebb szerepe. Az ISSO néhány felelőssége: • A biztonsági irányelv létrehozása és karbantartása • Felhasználók jelszavainak beállítása • Hálózatkonfiguráció • Eszközadminisztráció
pkgadm	Szoftvercsomag adminisztrátor. A pkgadm szerep felelős a rendszeren telepített szoftverekért, és alapértelmezésben jogosult a rendszerszoftverek telepítésére, frissítésére és eltávolítására.
sa	Rendszeradminisztrátor. Az SA szerep a napi adminisztrációhoz biztosítja a funkcionalitást és a következőkért felelős: • Felhasználóadminisztráció (kivéve a jelszóbeállítást) • Fájlrendszer-adminisztráció • Szoftvertelepítés frissítése • Hálózati démonkezelés • Eszközkiosztás

Szerepnév	Szerepleírás
secadm	Biztonsági adminisztrátor. A secadm szerep tartja karban a rendszer biztonsági beállításait. A secadm rendeli hozzá a felhasználókhoz az olyan attribútumokat, mint például a csoporttagságok, a szerepek, a jogosultságok és az engedélyek, valamint hozzárendeli azokat a szerepeket, amelyek korábban még nem voltak megadva. A secadm szerep rendeli hozzá ezenkívül a biztonsági attribútumokat a rendszerobjektumokhoz, az RBAC beállításokat, a hozzáférés-felügyeleti listákat, a tulajdonjogot és a tagságot is beleértve. A secadm szerep felelősségébe tartozik többek között: • Jelszavak hozzárendelése az új felhasználói fiókokhoz • Zárolt fiókok feloldása
so	Rendszeroperátor. Az SO szerep a napi műveletekhez biztosítja a funkcionalitást és a következőkért felelős: • Rendszer leállítása és újraindítása • Fájlrendszer biztonsági mentése, visszaállítása és kvóták kezelése • Rendszerhiba-naplózás, -nyomkövetés és statisztika • Terhelésadminisztráció
svcadm	Szolgáltatásadminisztrátor. A svcadm szerep engedélyezi, konfigurálja és letiltja a rendszerszolgáltatásokat. A szerep lehetővé teszi a hálózati attribútumok konfigurálását (például IP címek, útvonalak, hosztnevek és tűzfal irányelvek).
sysop	Rendszeroperátor. A sysop szerep tartja karban a teljes rendszer engedélyeit, amely tartalmazza a rendszerdiagnosztika futtatását és a rutin rendszerkarbantartás elvégzését is. A sysop feladatai többek között: • Naplófájlok és nyomtatási sorok kiürítése • Rendszerek leállítása és újraindítása
useradm	Felhasználóadminisztrátor. A useradm szerep a felhasználókarbantartással kapcsolatos magasabb szintű feladatokért felel, a jelszavak kezelése nélkül. A useradm az alapértelmezett biztonsági beállítások által meghatározott felhasználói fiókokat hozhatja létre, módosíthatja vagy törölheti. A szerep további szerepeket és csoportokat is létrehozhat alapértelmezett biztonsági beállításokkal.

Szerepátállítás:

Ha egy AIX Version 6.1 előtti AIX rendszert AIX rendszer kiterjesztett RBAC szintre frissít átállításos telepítésen keresztül, akkor az /etc/security/roles fájl átállítása megpróbálja frissíteni a fájlt az új funkcióhoz, az aktuális szerepképességek fenntartása mellett.

A fájlban lévő szerepmeghatározások megmaradnak és egyszerűen módosításra kerülnek, hogy egyedi szerepazonosítót tartalmazzanak a szerep megfelelő működése érdekében az új keretrendszerben. Az /etc/security/roles fájlban nem ismert előre meghatározott felhatalmazásokat a rendszer felhasználó által megadott felhatalmazásoknak tekinti. Az átállítás során ezek a felhatalmazásnevek bejegyzésként hozzáadásra kerülnek a helyi /etc/security/authorizations felhatalmazási adatbázishoz. A régi szerepmeghatározások átállításán felül az új előre meghatározott szerepek hozzáfűzésre kerülnek a fájlhoz. Az átállítás után a rendszeradminisztrátornak ellenőriznie kell, hogy a felhatalmazások a környezet által igényelt módon vannak-e megadva.

RBAC jogosultságok:

A kiterjesztett RBAC keretrendszer nagyban a rendszerjogosultságokra támaszkodik annak lehetővé tételében, hogy a nem jogosult felhasználók privilegizált feladatokat hajtsanak végre. A jogosultság a folytamat számára megnövelt funkcionalitás biztosításának mechanizmusa a rendszerhívásokban.

A jogosultságok alapja elsősorban egy kernelszintű szerkezet, mivel a meghatározás és az ellenőrzés többsége a kernelben történik. Felhasználószintű felületek jogosultságok parancsokhoz, eszközökhöz és folyamatokhoz rendelésének kezeléséhez biztosítottak.

Fontos megfigyelni a jogosultságok és felhatalmazások közötti különbséget. A jogosultságok és felhatalmazások szabályozzák a rendszerbiztonsági irányelv bizonyos megengedett kivételeit. A jogosultságok és felhatalmazások közötti különbség az adott folyamatokhoz rendelt jogosultságok, mialatt felhatalmazások vannak a felhasználókhoz rendelve szerepeken keresztül. Felhatalmazások tartoznak a szerephez és a szerepet birtokló felhasználóhoz, és ezek nem függenek a futtatandó programtól. A programhoz tartoznak jogosultságok és biztosítják a mechanizmust a rendszerbiztonsági irányelv finomhangolása érdekében. Ezekkel a hozzárendelt jogosultságokkal a folyamat végre tudja hajtani a kapcsoló privilegizált műveletet.

A jogosultságok az AIX kernelben a bitmaszk egyedi bitjeiként vannak megadva, amelyek kikényszerítik a privilegizált műveletek hozzáférés-vezérlését. Az AIX rendszerhez több mint 100 biztosított, ezáltal biztosítva a privilegizált műveletek nagyon finoman szabályozott vezérlését. A hozzáférés meghatározásakor egy rendszerhívásban a kernel meghatározza, hogy a folyamat rendelkezik-e a szükséges hozzárendelt jogosultságbittel, majd biztosítja vagy megtagadja a kérést.

A parancsmeghívásokhoz a privilegizált parancsadatbázison keresztül jogosultságok vannak rendelve, és a jogosultságok szabályozzák az eszközök elérését a privilegizált eszközadatbázison keresztül.

Jogosultság elnevezése és hierarchia:

Az AIX jogosultságokat a rendszeradminisztrátor nem hozhatja létre, nem módosíthatja és nem törölheti.

A rendelkezésre álló jogosultságok listája és a jogosultság rövid leírása a rendszeren a következő parancs futtatásával jeleníthető meg:

```
lspriv -v
```

Az AIX rendszeren biztosított jogosultságok listáját az AIX jogosultságok rész tartalmazza. Minden AIX jogosultság rendelkezik a jogosultság bit szöveges ábrázolásával, amely **PV_** karaktersorozattal kezdődik. A **PV_** előtag után használt elnevezési megállapodás a jogosultságok közötti hierarchikus kapcsolatot jelöli. A **PV_AU_** megfigyelési jogosultság például a **PV_AU_ADD**, **PV_AU_ADMIN**, **PV_AU_READ**, **PV_AU_WRITE** és **PV_AU_PROC** jogosultság szülője. A jogosultság ellenőrzésekor a rendszer először meghatározza, hogy a folyamat rendelkezik-e a legkisebb szükséges jogosultsággal, majd folytatja az ellenőrzést a hierarchiában felfele erősebb jogosultságokat keresve. A **PV_ROOT** jogosultság speciális jogosultság, amely az összes jogosultság szülőjét ábrázolja, a **PV_SU_** kivételével. A **PV_ROOT** jogosultságot hozzárendelő folyamat úgy viselkedik, mintha hozzárendelt volna minden jogosultságot a rendszeren a **PV_SU_** kivételével.

Folyamat jogosultsághalmazai:

Több jogosultsághalmaz van megadva a kernelben a privilegizált műveletek különböző vezérlésének biztosítása érdekében. A több jogosultsághalmaz lehetővé teszi az operációs rendszer számára dinamikus jogosultságvezérlés foganatosítását, az alkalmazások számára pedig a legkevesebb jogosultság elv kezelését.

A jogosultságok a folyamathoz a következő jogosultsághalmazokon keresztül vannak hozzárendelve:

Korlátozó jogosultsághalmaz (LPS)
: Megadja egy adott folyamat jogosultságainak kemény korlátját. A rendszerben egy jogosultságkiterjesztés aem növelheti a folyamatjogosultságokat ezen értéken túlra. Ez azt jelenti, hogy a folyamat a megadott

rendszerfelületek egyikével sem tud ennél az értéknél több jogosultságot elérni. Más szavakkal a folyamat mindig ezekre a jogosultságokra van korlátozva. Ez azt is jelenti, hogy a további jogosultsághalmazok mindig az LPS részhalmazai lesznek. Az LPS nem terjeszthető ki, de minden folyamat csökkentheti az LPS-t. Az LPS azonban csökkentés után nem állítható vissza az eredeti értékre. Az LPS csökkentésével a folyamat korlátozhatja a társított jogosultságok határait. A folyamat például csökkentheti a LPS-t egy egyéni felhasználó által biztosított program futtatása előtt. Alapértelmezésben a rendszeren elérhető minden jogosultság beállításra kerül az LPS-ben egy folyamathoz.

Maximális jogosultsághalmaz (MPS)
A folyamat által használható jogosultságok teljes halmaza. Az MPS tetszőleges LPS-ben lévő jogosultságot tartalmazhat, de az LPS-t nem haladhatja meg. Az MPS a folyamat élettartama során különböző okok miatt változhat. A lehetséges okok közül néhány:

- Az aktuális folyamat másik privilegizált parancsot hajt végre és további kapcsolódó jogosultságokat kap
- Ha a folyamat rendelkezik a megfelelő jogosultsággal, akkor ez kiterjesztheti az MPS-t programozottan, dinamikus módon

Hatályos jogosultsághalmaz (EPS)
A folyamathoz jelenleg aktív jogosultságok listája. Az EPS mindig a folyamat MPS-ének részhalmaza és a kernel használja a privilegizált műveletek hozzáférés-ellenőrzések végrehajtására. Az EPS-t a folyamat módosíthatja. Az EPS az MPS-sel egyenlővé tehető, de nem haladhatja meg azt. A folyamat végrehajthatja az EPS dinamikus módosítását a legkevesebb jogosultság elv kikényszerítése érdekében. A felhasználói tárterület kód például növelheti az EPS-ben lévő megfigyelési jogosultság bitet a **priv_raise** API segítségével a megfigyeléssel kapcsolatos rendszer- vagy kernelhívás előtt. A jogosultság a **priv_lower** API segítségével csökkenthető a megfigyelési hívás visszatérése után.

Örökölhető jogosultsághalmaz (IPS)
A szülőfolyamat által a leszármazott folyamat MPS-ének és EPS-ének átadott jogosultságok. Az IPS az LPS-ben lévő bármely jogosultságot tartalmazhatja, de nem haladhatja meg azt. Az IPS a folyamatban a következő módon állítható be:

- Ha a folyamat megfelelő jogosultsággal rendelkezik, akkor a **setppriv** rendszerhívással programozott módon kiterjesztheti az IPS-t
- Privilegizált parancs futtatása esetén a parancshoz rendelt **inheritprivs** attribútumban megadott jogosultságok hozzárendelésre kerülnek az IPS-hez.

Használt jogosultsághalmaz (UPS)
A folyamat élete során a hozzáférés-ellenőrzésekhez használt jogosultságokat jelöli. Az UPS segítségével meghatározhatók a folyamat számára szükséges jogosultságok. Amikor a kernel ellenőrzi, hogy a folyamat rendelkezik-e egy adott jogosultsággal, akkor egy sikeres ellenőrzést tárolt az UPS-ben a jogosultsághoz.

Munkapartíció jogosultsághalmaz (WPS)
A rendszer WPAR korlátozható, hogy ne engedélyezze a globális WPAR-ben engedélyezett összes privilegizált műveletet. A rendszer WPAR-ben engedélyezett privilegizált műveletek a WPS-sel szabályozhatók. A globális root korlátozott jogosultsághalmazt rendelhet a WPAR-hez a WPS segítségével. A WPS az /etc/wpar/secattrs konfigurációs fájlban vagy a WPAR indítása során a **/usr/sbin/startwpar** parancs segítségével adható meg. A WPAR-ben futó összes folyamat a WPS-sel megegyező LPS-sel rendelkezik.

A rendszeradminisztrátor az adminisztrációs parancsok segítségével listázhatja és módosíthatja a folyamat különböző jogosultsághalmazait. Az **lssecattr** parancs segítségével listázható az LPS, MPS, EPS, IPS és UPS. A **setsecattr** parancs segítségével módosítható az LPS, MPS, EPS és IPS. Az UPS a **setsecattr** paranccsal nem módosítható, mivel csak olvasható attribútum.

Privilegizált parancsadatbázis:

A felhatalmazások, szerepek és jogosultságok lehetővé teszik részletes biztonsági felügyelet megvalósítását. Azonban az RBAC kihasználata különféle rendszerműveletekkel lehetővé teszi RBAC biztonsági irányelv betartatását.

Míg történetileg egyes AIX parancsok felhatalmazása közvetlenül kerül ellenőrzésre, magát a végrehajtható kódot is módosítani kell ezen ellenőrzések végrehajtásához. A kiterjesztett RBAC mód a rendszer végrehajtható fájljainak módosítása nélkül biztosít egy keretrendszert a felhatalmazásellenőrzések kikényszerítésére és a társított jogosultságok megadására a privilegizált parancsadatbázison keresztül.

A privilegizált parancsadatbázis hozzáférést és jogosultságot ad a felhasználóknak olyan parancsokhoz, amelyeket egyébként nem lennének képesek futtatni, vagy amelyekkel megfelelő jogosultság hiányában nem lennének képesek elvégezni feladatukat. Az adatbázis elmenti az adott parancsok felhatalmazásinformációit, valamint a folyamat által a sikeres felhatalmazásellenőrzés esetén megkapott jogosultságokat. Ha az adatbázis helyileg van tárolva, akkor az az /etc/security/privcmds fájlban található és parancs-biztonsági attribútumok formájában tartalmaz információszakaszokat. Alább látható az adatbázis néhány alapvető attribútuma (az attribútumok teljes leírásáért tekintse meg az /etc/security/privcmds fájlt).

accessauths
: Felsorolja a parancs végrehajtását védő hozzáférési felhatalmazásokat. A felsorolt jogosultságok bármelyikével rendelkező felhasználó végrehajthatja a parancsot és a parancs által tartalmazott privilegizált műveletek közül néhányat vagy mindet.

innateprivs
: Az öröklött jogosultságokat a folyamat akkor kapja meg, ha a hívó átmegy a felhatalmazásellenőrzéseken.

authprivs
: A felhatalmazott jogosultságok olyan további jogosultságok, amelyeket a folyamat akkor kap meg, ha a felhasználó rendelkezik a társított felhatalmazással. Ez az attribútum a parancs alaposabb felügyeletét teszi lehetővé, megengedve felhasználók korlátozott halmazának további privilegizált műveletek végrehajtását.

inheritprivs
: Az örökölhető jogosultságokat a folyamat továbbadja az utódfolyamatainak.

secflags
: A biztonsági jelzők listája. Az FSF_EPS jelző hatására a parancs futtatásakor a maximális jogosultsághalmaz (MPS) betöltésre kerül a hatályos jogosultsághalmazba (EPS).

Ha egy kiterjesztett RBAC módú rendszer felhasználója megkísérel futtatni egy parancsot, a rendszer először a privilegizált parancsadatbázisban keresi meg a parancsot. Ha a parancs létezik az adatbázisban, akkor ellenőrzésre kerülnek a felhasználó munkamenetéhez tartozó jogosultságok, valamint a parancshoz tartozó **accessauths** attribútum. Ha a munkamenet rendelkezik a felsorolt jogosultságok egyikével, akkor a felhasználó futtathatja a parancsot tekintet nélkül arra, hogy a felhasználó átmegy-e a parancs DAC végrehajtási ellenőrzésein. Meghíváskor a parancs folyamata a maximális privilégiumhalmazához (MPS) rendelt **innateprivs** attribútumban felsorolt jogosultságokkal rendelkezik. További felhatalmazásellenőrzések kerülnek végrehajtásra az **authprivs** attribútumban felsorolt felhatalmazás-jogosultság párokra. Ha a munkamenet rendelkezik a felsorolt felhatalmazások egyikével, akkor a társított jogosultságok is hozzáadásra kerülnek a parancs folyamatának maximális privilégiumhalmazához. A privilegizált parancsadatbázis azon parancsbejegyzése, amely **secflags** attribútumában be van állítva az **FSF_EPS** érték, a parancs meghívásakor a maximális privilégiumhalmaz összes jogosultságát a hatályos privilégiumhalmazhoz (EPS) rendeli.

A parancs privilegizált, ha a privilegizált parancsadatbázis tartalmazza. Noha az adatbázisban nem szereplő setuid programok technikailag privilegizált parancsok, az RBAC viselkedés leírásakor nem nevezhetők privilegizált parancsoknak. Ha egy parancshoz nem tartozik bejegyzés a privilegizált parancsadatbázisban, akkor az nem privilegizált parancs és a hozzáférést a DAC és maga a parancs tartatja be. Ezen kívül ha egy parancs a privilegizált parancsadatbázisban van felsorolva, de a felhasználó munkamenete nem rendelkezik a parancs meghívását lehetővé tévő felhatalmazással, akkor a rendszer visszatér a DAC hozzáférés ellenőrzéséhez és lehetővé teszi a parancs futtatását, amennyiben ezek az ellenőrzések sikeresek.

Számos felügyeleti parancs került létrehozásra a privilegizált parancsadatbázis kezelése és lekérdezése érdekében. A privilegizált parancs adatbázisban lévő bejegyzések a **setsecattr** paranccsal létrehozhatók vagy módosíthatók, az **lssecattr** paranccsal megjeleníthetők, az **rmsecattr** paranccsal pedig eltávolíthatók.

Parancs szükséges jogosultságainak meghatározása:

Sok rendszeradminisztrátori alkalmazás számára jogosultság szükséges a megfelelő működéshez. Míg a privilegizált parancsadatbázis tartalmazza az előre meghatározott parancsok egy halmazát, a rendszeradminisztrátoroknak szükségük lehet az adott környezetre jellemző bejegyzések felvételére. A privilegizált parancsadatbázis lehetővé teszi bejegyzések hozzáadását az adatbázishoz. A parancs eléréséhez a megfelelő jogosultságot az **accessauths** attribútumban kell felsorolni.

Kétféle módon lehet használni és ellenőrizni felhatalmazást az AIX operációs rendszeren a kiterjesztett RBAC keretrendszerrel:

- Access Auths (Hozzáférési jogosultság): A privilegizált parancsadatbázisban megadott attribútum, amely a jogosultságok neveit vesszővel elválasztott listában tartalmazza. A parancs futtatására az a felhasználó jogosult, akinek a jelenlegi szekciója a listában felsorolt jogosultsággal bír. Ezt a rendszerbetöltő ellenőrzi védett privilegizált végrehajtható fájlok futtatásakor.
- Check Auths (checkauths()): A checkauths() API segítségével bizonyos jogosultság vagy jogosultsági lista ellenőrizhető egy programmal. A megadott jogosultságokat a program az aktuális szekció szerepkörében lévő jogosultságokhoz viszonyítva ellenőrzi. Az ellenőrzés eredményének alapján a program végrehajthat privilegizált műveleteket.

Mielőtt a privilegizált parancsadatbázishoz adna egy parancsot, meg kell határozni a jogosultsági készleteket, hogy biztosított legyen a parancs végrehajtásának engedélyezése. Egy programnak vagy alkalmazásnak további jogosultsági ellenőrzést kell belsőleg végeznie. Meg kell határozni egy folyamatban használt jogosultságok listáját, amelyek hozzárendelhetők egyéni szerepkör létrehozása során.

A parancs szükséges jogosultságainak meghatározására a következő az alapstratégia:

1. Rendelje a **PV_ROOT** jogosultságot a hívó parancsértelmezőhöz vagy használjon *aix* jogosultságú szerepkört.

   Fontos: Globális WPAR esetén a **PV_ROOT** jogosultságot egy hívó parancsértelmező folyamat hatályos és maximális jogosultságkészletéhez kell rendelni. Rendszer WPAR-on belül ezt a jogosultságot hozzá kell adni a folyamat örökölt jogosultságkészletéhez is.
2. Futtassa a parancsot.
3. Rögzítse a folyamathoz használt jogosultságokat.
4. Az *Access Auths* alatti jogosultságokat tárolja a privilegizált parancsadatbázis parancsának **accessauths** attribútumában. A rendszer szerepköreinek létrehozásakor használhatók a *Check Auths* alatti jogosultságok.

Ezeket a lépéseket felügyelt környezetben kell végrehajtani, mert a **PV_ROOT** jogosultságot egy parancsértelmező kapja meg, vagy az *aix* jogosultságú szerepkört használ, és mivel mindkét jogosultság különösen erős. Ezen kívül a parancs futtatása hatással lehet a rendszerre úgy, hogy az más felhasználókat is érinthet. A gyakorlatban ez valószínűleg egy ismételt közelítésen alapuló eljárás lesz. A jogosultságok teljes halmazának megszerzése érdekében a parancsot valószínűleg többször is le kell futtatni különböző beállításokkal és kapcsolókkal, és sokáig futó alkalmazások esetén ez hosszú ideig tarthat. A folyamat szükséges jogosultsághalmaza egyszerűen összegyűjthető a következő eljárások egyikének segítségével. Ezeket egy megfelelő jogosultságokkal rendelkező adminisztrátor hajthatja végre:

traceauth
: Paraméterként adja meg a végrehajtandó parancsot. A **traceauth** parancs futtatja a parancsot és rögzíti a folyamat élettartama során használt mindkét típusú jogosultságot. A parancs befejeződésekor a **traceauth** parancs a **szabványos kimeneten** megjeleníti a használt jogosultságokat.

lssecattr
: Ha a parancs egy hosszú ideig futó folyamat, akkor az **lssecattr** parancs segítségével jeleníthetők meg a folyamat által használt jogosultságok. A rendszer jogosultság-nyomkövetésének engedélyezéséhez futtassa a következő parancsot:

  setrunmode –c; setsecconf –o traceauth=enable Egy folyamat használt jogosultságának megjelenítéséhez futtassa az **lssecattr** parancsot a következőképp, behelyettesítve a megfigyelt folyamat azonosítóját (PID):

lssecattr –p –A *PID*

A szükséges jogosultságok meghatározása után a “Parancs hozzáadása a privilegizált parancsadatbázishoz” oldalszám: 95 részben leírtak szerint adja hozzá a parancsot a privilegizált parancsadatbázishoz. A parancsot ezután egy felhatalmazott felhasználóként kell futtatni a megfelelő működés ellenőrzése érdekében.

Parancs által igényelt jogosultságok meghatározása:

Sok alkalmazás számára bizonyos jogosultságok szükségesek a megfelelő végrehajtáshoz. Míg a privilegizált parancsadatbázis tartalmazza az előre meghatározott parancsok egy halmazát, a rendszeradminisztrátornak szüksége lehet az adott alkalmazásra és környezetére jellemző bejegyzések felvételére. A privilegizált parancsadatbázis lehetővé teszi bejegyzések hozzáadását parancsokhoz és a hozzájuk tartozó jogosultságokhoz.

A parancs privilegizált parancsadatbázishoz adása előtt meg kell határozni a szükséges jogosultságok minimális halmazát annak biztosítása érdekében, hogy a parancs végrehajtása a lehető legbiztonságosabb legyen. A megfelelő végrehajtáshoz szükségesen felül megadott jogosultságok sértik a legkevesebb jogosultság elvét. Emiatt a privilegizált parancs rendszerhez adásának fontos lépése a minimálisan szükséges jogosultságok meghatározása.

A parancs minimálisan szükséges jogosultságainak meghatározásának alapvető stratégiája a következő:

1. Az Information System Security Officer (ISSO) vagy az isso szerepkörrel rendelkező felhasználó hozzárendelheti a **PV_ROOT** jogosultságot a rendszeradminisztrátorhoz a privilegizált adatbázishoz hozzárendelendő parancs végrehajtásával. A **PV_ROOT** jogosultság hozzárendelése a meghívott parancsértelmezőhöz a setsecattr parancs segítségével lehetséges. Például:

   setsecattr -p eprivs=PV_ROOT mprivs=PV_ROOT $$
2. Futtassa a parancsot a jogosultságok készletének összegyűjtéséhez.
3. Rögzítse a folyamathoz használt jogosultsághalmazt.
4. A szükséges jogosultságokat tárolja a parancs **innateprivs** attribútumában a privilegizált parancsadatbázisban.

Ezeket a lépéseket felügyelt környezetben kell végrehajtani, mivel a **PV_ROOT** jogosultságot egy parancsértelmező kapja meg, a **PV_ROOT** jogosultság pedig különösen erős. Ezen kívül a parancs futtatása hatással lehet a rendszerre úgy, hogy az más felhasználókat is érinthet. A gyakorlatban ez valószínűleg egy ismételt közelítésen alapuló eljárás lesz. A jogosultságok teljes halmazának megszerzése érdekében a parancsot valószínűleg többször is le kell futtatni különböző jelzőkkel és kapcsolókkal, és sokáig futó alkalmazások esetén ez hosszú ideig tarthat. A folyamat szükséges jogosultsághalmaza egyszerűen összegyűjthető a következő eljárások egyikének segítségével. Ezeket egy megfelelő jogosultságokkal rendelkező adminisztrátor hajthatja végre:

tracepriv
: Egy argumentumot vár, ami a végrehajtandó parancs. A **tracepriv** parancs futtatja a parancsot és rögzíti a folyamat élettartama során használt jogosultságokat. A parancs befejeződésekor a **tracepriv** parancs a **szabványos kimeneten** megjeleníti a használt jogosultságokat.

lssecattr
: Ha a parancs egy hosszú ideig futó folyamat, akkor az **lssecattr** parancs segítségével jeleníthetők meg a folyamat által használt jogosultságok. A folyamathoz használt jogosultsághalmaz megjelenítéséhez futtassa a következő parancsot, a megfigyelendő folyamat folyamatazonosítójának helyettesítésével:

  lssecattr –p –a uprivs *PID*

A minimálisan szükséges jogosultságok meghatározása után hajtsa végre a “Parancs hozzáadása a privilegizált parancsadatbázishoz” oldalszám: 95 részben leírtakat a parancs privilegizált parancsadatbázishoz adása érdekében. A parancsot ezután egy felhatalmazott felhasználóként kell futtatni a megfelelő működés ellenőrzése érdekében.

Jogosultságkiterjesztés:

Ha a **fork** rendszerhívás új folyamatot hoz létre, akkor a **fork** ugyanazokat a jogosultságokat biztosítja a folyamatnak, mint a szülőfolyamatnak (a **fork** rendszerhívás által meghívott folyamat). Ha a folyamat **exec** rendszerhívást hajt végre

egy végrehajtható fájlon, akkor az **exec** újraszámítja a jogosultságokat a végrehajtható fájlhoz az **exec** és a végrehajtható fájl által jelenleg birtokolt jogosultságok alapján.

A kiterjesztett jogosultságok a következők szerint kerülnek kiszámításra:

1. Először a régi (szülő) folyamat által birtokolt örökölhető jogosultságok és a végrehajtható fájl által birtokolt belső jogosultságok uniója (bit szintű VAGY művelet) kerül kiszámításra.
2. Ha a felhasználó megfelelően felhatalmazott, akkor az előző lépés eredményének és a felhatalmazott jogosultságok uniója (bit szintű VAGY) kerül kiszámításra.
3. Korlátozó jogosultságok esetén az előző lépés eredményének és a korlátozó jogosultságoknak a metszete kerül kiszámításra. , Amennyiben vannak, a korlátozó jogosultságok az **exec** rendszerhíváson keresztül öröklődnek.
4. Az unió által eredményezett felhatalmazások halmaza lesz az új maximális jogosultsághalmaza.
5. Ha az örökölt jogosultságok léteznek a végrehajtható fájlban, akkor hozzárendelésre kerülnek az új folyamat örökölhető jogosultsághalmazához. Ellenkező esetben a régi (szülő) folyamat által birtokolt örökölhető jogosultságok átadásra kerülnek az új folyamat örökölhető jogosultsághalmazába.

Ha a végrehajtható fájl rendelkezik beállított **FSF_EPS** fájlbiztonsági jelzővel, akkor az új folyamat hatályos jogosultságainak halmaza megegyezik a maximális jogosultsághalmazzal. Ellenkező esetben az új folyamat hatályos jogosultságai a régi (szülő) folyamat által birtokolt örökölhető jogosultságokkal egyeznek meg.

Parancs hozzáadása a privilegizált parancsadatbázishoz:

Alaposan meg kell gondolnia, hogy hozzáad-e egy parancsot a privilegizált parancsadatbázishoz a megfelelő felhatalmazások és jogosultságok hozzárendelésének biztosítása érdekében.

A parancshoz érvényes attribútumok teljes leírásáért tekintse meg az /etc/security/privcmds fájlt. A következő kérdések útmutatóként használhatók annak meghatározásához, hogy szükséges-e bejegyzés a parancshoz:

1. Szükség van felhatalmazási hozzáférés-felügyeletre a parancs futtatásához?

   IGEN Ha a felhatalmazás nem létezik, akkor hozza létre az **mkauth** parancs segítségével. Adja meg a felhatalmazást az **accessauths** attribútumban.

   NEM Ha minden felhasználó futtathatja a parancsot, akkor az **accessauths** attribútumban adja meg az **ALLOW_ALL** felhatalmazást.
2. A parancs tulajdonosának vagy csoportjának engedélyezni kell a parancs futtatását akkor is, ha nem rendelkeznek megfelelő felhatalmazással?

   IGEN Vegye fel az **ALLOW_OWNER** vagy **ALLOW_GROUP** felhatalmazást az **accessauths** attribútumban a felhatalmazások listájára.
3. A parancs végrehajtásakor igényli felhatalmazások egy adott halmazát?

   IGEN Futtassa a parancsot különféle kapcsolókkal root felhasználóként a **tracepriv** paranccsal az **innateprivs** attribútum által igényelt jogosultságok meghatározása érdekében.
4. Az adott jogosultsággal rendelkező felhasználóknak meg kell kapniuk további jogosultságokat is?

   IGEN Adja meg a további felhatalmazás-jogosultság párokat az **authprivs** attribútumban.
5. A parancsnak SUID vagy SGID programként kell viselkednie?

   IGEN Adja meg az EUID vagy EGID egyikét, értelemszerűen.
6. A parancshoz rendelt jogosultságokat át kell adni a leszármazott folyamatoknak?

   IGEN Adja meg a jogosultságokat az **inheritprivs** attribútumban.
7. A parancs hatályos privilégiumhalmazának meg kell egyeznie a maximális privilégiumhalmazzal a parancs meghívásakor?

   IGEN Adja meg az **FSF_EPS** jelzőt a secflags attribútumhoz.

   NEM Ne adja meg a **secflags** attribútumot. A parancskód növeli és csökkenti a jogosultságokat, ha az **FSF_EPS** jelző nincs megadva.

8. A parancsot a speciális 0 valós felhasználói azonosítóval kell futtatni?

   IGEN Adja meg a RUID attribútumot.

9. A parancs nagyon fontos és meghívásához több személy jelenléte is szükséges?

   IGEN Adja meg az **authroles** attribútumot és adja meg a szerepkörök listáját tartalmazó értéket. A parancs végrehajtása előtt az egyes szerepkörök felhasználóit hitelesíteni kell.

A fenti kérdések megválaszolása után futtassa a **setsecattr** parancsot a megfelelő paraméterekkel a parancs adatbázishoz adásához. Ha a parancs létező SUID vagy SGID parancs, akkor meg kell fontolni a **SUID** és **SGID** bitek eltávolítását a fájlból, így betartatható a legkevesebb jogosultság modell.

Privilegizált eszközadatbázis:

A privilegizált eszközadatbázis azon felhatalmazások listáját tárolja, amelyeknek engedélyezett egy eszköz olvasása vagy írása. Ez az adatbázis lehetőséget biztosít az adminisztrátornak az eszköz hozzáférésének a hagyományos eszközhozzáférés-felügyeleti eszközökkel kezelhetőnél részletesebb felügyeletére.

Az adatbázist helyi tárolás esetén az /etc/security/privdevs fájl tartalmazza. Az adatbázis a következő attribútumokban tárolja az adott eszköz olvasási vagy írási műveletekhez történő eléréséhez szükséges jogosultságokat:

readprivs
: Felsorolja azokat a jogosultságokat, amelyek számára engedélyezett az eszköz olvasása

writeprivs
: Felsorolja azokat a jogosultságokat, amelyek számára engedélyezett az eszköz írása

Privilegizált eszköz olvasási módban történő megnyitására tett kísérlet esetén a megnyitás csak akkor engedélyezett, ha a **readprivs** attribútumban megadott jogosultságok egyike létezik a folyamat hatályos jogosultsághalmazában (EPS). Hasonlóképpen, az eszköz írási módban történő megnyitásakor a **writeprivs** attribútum egyik jogosultságának léteznie kell az EPS-ben.

Az eszközök privilegizált eszközadatbázishoz adásának folyamata nem gyakori művelet. Az **lssecattr** és **setsecattr** parancsok segítségével kiíratható és kezelhető az adatbázis, de bejegyzések hozzáadása az adatbázisba vagy azok módosítása alapos vizsgálatot igényel. Mivel az eszköz olvasási és írási engedélye jogosultságokon keresztül vezérelhető, a megfelelő jogosultságok megadásának biztosítása érdekében az eszközhozzáférést igénylő parancsok és alkalmazások alapos vizsgálatára van szükség.

Privilegizált fájladatbázis:

A hagyományos UNIX rendszereken számos rendszerkonfigurációs fájlt a root felhasználó birtokol és mások nem módosíthatják közvetlenül. A RBAC lehetővé teszi, hogy a felhasználó módosítsa a rendszerkonfigurációs fájlt azáltal, hogy aktivál egy szerepet és lefuttat egy parancsot a fájl módosításához szükséges jogosultságok megszerzése érdekében.

Néhány AIX konfigurációs fájl nem rendelkezik parancsfelülettel a fájl módosításának lehetővé tételéhez. Ezekben az esetekben szükség van egy eszközre, amely lehetővé teszi, hogy a megfelelő felhatalmazással rendelkező adminisztrátor közvetlenül szerkessze és mentse a fájlt, amelyhez más esetben nem férhetne hozzá.

A privilegizált fájladatbázis lehetővé teszi a felhatalmazások használatát a rendszerkonfigurációs fájl elérésének meghatározása érdekében. Az adatbázis helyi tárolása esetén ez az /etc/security/privfiles fájlban található. Ez az adatbázis leképezi a konfigurációs fájlokat a fájlok megjelenítéséhez és módosításához szükséges felhatalmazásokra. A konfigurációs fájl elérését az adatbázisban a következő attribútumok szabályozzák:

readauths
: A fájl olvasását engedélyező felhatalmazások listája

writeauths
: A fájl írását engedélyező felhatalmazások listája (ebben az esetben olvasás felhatalmazást magában foglalja)

A privilegizált fájladatbázisban lévő bejegyzések az **lssecattr** parancs segítségével listázhatók, a **setsecattr** parancs segítségével pedig létrehozhatók és módosíthatók. A privilegizált fájladatbázisban megadott fájlokat a felhatalmazott felhasználók a **/usr/bin/pvi** paranccsal elérhetik. A **pvi** parancs a **vi** szerkesztő **/usr/bin/tvi** parancs alapján privilegizált és korlátozott változata. A **pvi** parancsra érvényesek ugyanazok a biztonsági óvintézkedések, mint a **tvi** parancsra (például nincs **–r** és **-t** kapcsoló, nincsenek parancsértelmező-kilépések, nincsenek felhasználó által megadott makrók), de ezen felül a következő korlátozásokat is kikényszeríti:

- A rendszernek kiterjesztett RBAC módban kell lennie.
- Csak a privilegizált adatbázisban megadott fájlok nyithatók meg.
- Egyszerre csak egy fájl nyitható meg.
- A parancssorban megadottól eltérő fájlnévbe írás nem megengedett.
- Az /etc/security/privfiles fájl a **pvi** paranccsal nem szerkeszthető.
- A hivatkozások megnyitására tett kísérletek meghiúsulnak. Csak a szabályos fájlok szerkeszthetők.

A felhatalmazásellenőrzések a fájl megnyitása előtt kerülnek végrehajtásra. Ha a felhatalmazás megfelelő, akkor a folyamat jogosultsághalmaza kibővítésre kerül a **PV_DAC_R** vagy **PV_DAC_W** tartalmazása érdekében (attól függően, hogy a fájl megnyitásra kerül-e olvasásra vagy írásra). Ha a felhatalmazás nem megfelelő, akkor hibaüzenet jelenik meg és a felhasználó nem férhet hozzá a fájlhoz a **pvi** paranccsal.

Kernel biztonsági táblázatok:

A felhatalmazási, szerep-, privilegizált parancs-, és privilegizált eszközadatbázisokban található információk nem kerülnek felhasználásra biztonsági szempontokhoz az adatok betöltéséig a kernel által kernel biztonsági táblázatok (KST) néven kijelölt területre. Kiterjesztett RBAC módban a felhatalmazás- és jogosultságellenőrzések a kernelben kerülnek végrehajtásra, ezért az adatbázisokat használat előtt el kell küldeni a kernelnek.

A KST a következő résztáblázatokból áll:

- Kernel felhatalmazási táblázat (KAT)
- Kernel szereptáblázat (KRT)
- Kernel parancstáblázat (KCT)
- Kernel eszköztáblázat (KDT)

Az összes táblázat vagy a kiválasztott táblázatok a felhasználói szintről a **setkst** paranccsal küldhetők el a kernelnek. A KRT és KCT a KAT-tól függenek, így a KAT frissítésre való kiválasztásakor a KRT és a KCT is frissítésre kerül annak ellenőrzése érdekében, hogy a táblázatok szinkronizálva vannak-e. A KST frissítésének előnyben részesített módja az összes szükséges adatbázis létrehozása vagy módosítása a felhasználói szinten (például az **mkauth**, **chauth**, **mkrole** és **setsecattr** parancsokkal), majd elküldése a kernelnek a **setkst** parancs segítségével. A táblázatok kernelbe betöltése után az **lskst** parancs segítségével megjeleníthetők az egyes táblázatok által tartalmazott információk.

A KST adott táblázata mindig teljes táblázatként kerül elküldésre. Más szóval a KST nem engedélyezi az egyedi bejegyzések módosítását, a teljes táblázatot le kell cserélni. A táblázatok kernelbe küldése előtt a **setkst** parancs ellenőrzi a táblázatokat és a köztük fennálló kapcsolatokat. A **setkst** parancs az inittab fájlban is el van helyezve annak biztosítása érdekében, hogy az adatbázisok a rendszerbetöltési folyamat elején elküldésre kerülnek a KST-nek.

Ha valamilyen okból a táblázatok nem hozhatók létre vagy nem tölthetők be a kernelbe, és korábban nem kerültek táblázatok betöltésre, akkor a rendszer úgy működik, mintha nem lennének felhatalmazások vagy szerepek. Ebben a szituációban a felhatalmazás- és szerepellenőrzést igénylő parancsok, alkalmazás programozási felületek és rendszerhívások hibával térnek vissza, mivel nem található egyezés. A rendszer működése ebben az állapotban nagyon hasonló az örökölt RBAC módhoz, azzal a kivétellel, hogy a felhasználók nem érhetik el a parancsok felhatalmazást kikényszerítő kódrészeit.

A root felhasználó letiltása:

Kiterjesztett RBAC módban beállítható a rendszer úgy, hogy a root felhasználó ne kapjon speciális hatáskört és a rendszer normál felhasználóként kezelje.

Történetileg a root felhasználó 0 felhasználói azonosítóját az operációs rendszer privilegizált azonosítóként kezelte és engedélyezte a betartatott biztonsági ellenőrzések kihagyását. A root felhasználó letiltása hatékonyan eltávolítja az operációs rendszer azon ellenőrzéseit, amelyek lehetővé teszik a 0 felhasználói azonosítójú felhasználónak a biztonsági ellenőrzések kihagyását, ehelyett megköveteli, hogy a sikeres biztonsági ellenőrzés érdekében a folyamat rendelkezzen felhatalmazással. A root felhasználó letiltása minimalizálja a támadók által okozható kárt, mivel megszünteti a rendszerben az egyetlen, minden hatalommal rendelkező felhasználót. A root felhasználó letiltása után a rendszeradminisztrációt privilegizált szerepet kapott felhasználóknak kell végrehajtaniuk.

A root hatásköre az **/usr/sbin/setsecconf** parancs segítségével tiltható le. A root felhasználó hatáskörének letiltásához futtassa a következő parancsot, majd indítsa újra a rendszert:

```
setsecconf –o root=disable
```

A parancs futtatása után a root felhasználói fiók nem lesz elérhető távoli vagy helyi bejelentkezés, illetve a su parancs használatával. Mivel azonban a root felhasználói fiók marad a fájlrendszer fájljainak tulajdonosa, a fiók újraengedélyezése után a felhasználónak hozzáféréssel rendelkezik a privilegizált fájlokhoz.

A rendszeren a root felhasználói fiók letiltása után a root által birtokolt folyamatok nem rendelkeznek többé speciális hatáskörrel vagy jogosultságokkal. Ezt úgy lehet elképzelni, mintha a rendszeren a root által birtokolt setuid alkalmazások nem kerültek volna hozzáadásra a privilegizált parancsadatbázishoz. Ezek a setuid alkalmazások valószínűleg hibásan fognak működni a root nélküli környezetben, mivel nem képesek végrehajtani a privilegizált műveleteket. Root nélküli rendszeren az összes privilegizált műveletet végrehajtó parancsot hozzá kell adni a privilegizált parancsadatbázishoz és megfelelő jogosultságot kell hozzájuk rendelni. Emiatt a rendszert és a rajta használt alkalmazásokat alapos elemzésnek kell alávetni a root felhasználó hatáskörének letiltása előtt.

Távoli RBAC adatbázis támogatása:

Vállalati környezetben szükség van az általános biztonsági irányelv megvalósításának és kikényszerítésének képességére a környezet minden rendszerén. Ha az irányelvet vezérlő adatbázisok minden rendszeren függetlenül kerülnek tárolásra, akkor a biztonsági irányelv kezelése a kijelölt rendszeradminisztrátor számára terhes lehet. Az AIX kiterjesztett RBAC mód lehetővé teszi az RBAC adatbázisok LDAP-ben való társítását, így a rendszer összes környezetének biztonsági irányelve központilag kezelhető.

Az AIX rendszerben támogatás biztosított az LDAP-ben tárolni kívánt összes RBAC-vel kapcsolatos adatbázishoz. A következők az RBAC-vel kapcsolatos adatbázisok:

- Felhatalmazási adatbázis
- Szerepadatbázis
- Privilegizált parancsadatbázis
- Privilegizált eszközadatbázis
- Privilegizált fájladatbázis

Megjegyzés: Az LDAP-ben tárolt felhatalmazási adatbázis csak felhasználó által megadott felhatalmazásokat tartalmaz. A rendszer által meghatározott felhatalmazások nem tárolhatók LDAP-ben és minden kliensrendszeren helyiek maradnak.

Az AIX segédprogramokat biztosít a helyi RBAC adatok LDAP-re való egyszerű exportálásához, a kliens beállításához LDAP-ben lévő RBAC adatok használatához, az RBAC adatok kikeresésének vezérléséhez és a kliensrendszer LDAP adatainak kezeléséhez. A következő részek a kiterjesztett RBAC-ben biztosított LDAP szolgáltatásokkal kapcsolatos további információkat tartalmaznak:

RBAC adatok exportálása LDAP címtárba:

Az LDAP RBAC adatbázis-lerakatként történő használatának kezdeti előkészületei megkövetelik az LDAP szerver feltöltését az RBAC adataival.

Az LDAP szerverre telepítve kell lennie az LDAP RBAC sémájának ahhoz, hogy az LDAP ügyfelek az RBAC adatokhoz használhassák a kiszolgálót. Az LDAP RBAC sémája az AIX rendszer /etc/security/ldap/sec.ldif fájljában érhető el. Az LDAP szerver sémáját az **ldapmodify** parancs segítségével ezzel a fájllal kell frissíteni.

A **/usr/sbin/rbactoldif** fájl segítségével beolvashatók a helyi RBAC adatbázisok adatai és kiírhatók az LDAP címtárnak megfelelő formátumban. Az **rbactoldif** parancs által előállított kimenet fájlba menthető, majd az az **ldapadd** parancs segítségével az LDAP szerver adatokkal történő feltöltésére használható. A helyi rendszer következő adatbázisait használhatja az **rbactoldif** parancs LDAP RBAC adatainak előállításához:

- /etc/security/authorizations
- /etc/security/privcmds
- /etc/security/privdevs
- /etc/security/privfiles
- /etc/security/roles

Az RBAC adatok LDAP tárterületének helyét át kell gondolni. Ajánlott az LDAP címtárban az RBAC adatokat a felhasználó- és csoportadatokkal egyező szülő megkülönböztetett név alatt elhelyezni. Az adatok hozzáférés-felügyeleti listáit ezután szükség szerint a kiválasztott biztonsági irányelvnek megfelelően kell beállítani.

LDAP klienskonfiguráció az RBAC-hoz:

A rendszert LDAP kliensként kell beállítani az LDAP-ban tárolt RBAC adatok használatához.

Az AIX **/usr/sbin/mksecldap** parancs segítségével beállíthatja a rendszert LDAP kliensként. Az **mksecldap** parancs dinamikusan keresi az LDAP szerveren a felhatalmazási, szerep-, privilegizált parancs, eszköz- és fájladatokat, az eredményeket pedig az /etc/security/ldap/ldap.cfg fájlba menti.

A rendszer az **mksecldap** parancs segítségével LDAP kliensként történő sikeres beállítása után a rendszeren további beállításokat kell végezni az LDAP engedélyezéséhez az RBAC adatok kikeresési tartományaként. Az /etc/nscontrol.conf fájlt módosítani kell, hogy az LDAP címtárban tárolt adatbázisok **secorder** attribútuma tartalmazza az LDAP-t.

A rendszer LDAP kliensként és RBAC adatok kikeresési tartományaként való beállítása után az **/usr/sbin/secldapclntd** kliensdémon rendszeres időközönként lekéri az RBAC adatokat az LDAP címtárból és az adatokat a **setkst** parancs használatával elküldi a kernel biztonsági táblázatoknak (KST). A démon által az RBAC adatok LDAP címtárból történő lekérésére használandó időköz az /etc/security/ldap/ldap.cfg fájl **rbacinterval** attribútumával állítható be. Az attribútum alapértelmezett értéke 3600, amely megadja, hogy az LDAP címtárból az RBAC adatokat óránként egyszer kell lekérni és a KST-t óránként kell frissíteni. A KST saját kezűleg is frissíthető, ha egy adminisztrátor a **setkst** parancsot futtatja.

Névszolgáltatás-vezérlőfájl:

Az RBAC adatok elhelyezkedhetnek szigorúan helyi fájlban, szigorúan LDAP-ban, vagy összefésülhetők a helyi fájlokban és az LDAP-ban az /etc/nscontrol.conf névszolgáltatás-vezérlőfájlban egy adott adatbázis konfigurálásával.

A felhatalmazás-, szerep-, privilegizált parancs-, eszköz- és fájladatbázisok keresési sorrendje egyenként meg van határozva az /etc/nscontrol.conf fájlban. Az adatbázisok keresési sorrendje a **secorder** attribútum segítségével van megadva, amely tartományok vesszőkkel elválasztott listája. Példa egy felhatalmazási adatbázis konfigurációjára:

```
authorizations:
        secorder = LDAP,files
```

Ez a példa megadja, hogy a felhatalmazási lekérdezéseknek először az LDAP-ban, majd a helyi fájlokban kell keresniük, ha a felhatalmazás nem található az LDAP-ban. A rendszer által elérhető felhatalmazások gyűjteménye az LDAP és a helyi fájlok által biztosított felhatalmazások összefésülésével kapott halmaz. Az összefésülés nem egyszerűen a két tartomány értékeinek kombinációja, hanem az értékek uniója. A fenti konfiguráció esetén minden LDAP felhatalmazás, majd a helyi fájlokból csak az egyedi felhatalmazások kerülnek hozzáadásra az eredményhez.

A módosításokra és törlésekre először az első felsorolt tartományban történik kísérlet, a soron következő tartományokban pedig csak akkor, ha az entitás nem található az első tartományban. Ebben az esetben a keresésre először az LDAP címtárban történik kísérlet, a helyi fájlokban pedig csak akkor, ha a felhatalmazás nem található az LDAP címtárban. Az új bejegyzések mindig a **secorder** attribútumban felsorolt első tartományban kerülnek létrehozásra. A fenti példában az új felhatalmazás létrehozása az LDAP adatbázisban történik meg.

Ha nincs adatbázis-bejegyzés az /etc/nscontrol.conf fájlban vagy a fájl nem létezik, akkor az adatbázis lekérdezései és módosításai csak a helyi fájladatbázisban kerülnek végrehajtásra. Az adatbázis konfigurációja a fájlban a **chsec** paranccsal állítható be és az **lssec** paranccsal sorolható fel. A felhatalmazási adatok először az LDAP címtárból, majd a helyi fájlokból történő lekérésének beállításához futtassa a következő parancsot:

```
chsec –f /etc/nscontrol.conf –s authorizations –a secorder=LDAP,files
```

Az /etc/nscontrol.conf fájlban található konfiguráció felügyeli mind a függvénytári, mind a parancssori felületet. Az alkalmazások egy adatbázis **secorder** attribútumának aktuális értékét a **getsecorder** felület használatával kérhetik le. A **secorder** attribútum értéke a **setsecorder** felület használatával bírálható felül egy folyamathoz.

RBAC parancsengedélyezési LDAP-hoz:

Az összes RBAC adatbázis-kezelési parancs engedélyezett az /etc/nscontrol.conf fájlban található konfiguráció használatához és az entitás lekérdezéséhez, módosításához, létrehozásához vagy eltávolításához az adott adatbázishoz meghatározott tartományokból.

Alapértelmezésben a tartományok az adatbázis **secorder** attribútumában meghatározott módon kerülnek feldolgozásra, de ez felülbírálható a **–R** parancssori kapcsoló használatával. A **–R** kapcsoló megadása a parancshoz kikényszeríti a művelet adott tartományon történő végrehajtását és felülbírálja az /etc/nscontrol.conf fájlban található konfigurációt. A következő RBAC adatbázis-kezelési parancsok engedélyezettek a távoli tartományok támogatásához:

- **mkauth**, **chauth**, **lsauth** és **rmauth**
- **mkrole**, **chrole**, **lsrole** és **rmrole**
- **setsecattr**, **lssecattr** és **rmsecattr**

Ezen kívül a **setkst** parancs is engedélyezett az/etc/nscontrol.conf fájl által tartalmazott konfiguráció használatához. A **setkst** parancs lekéri az adott adatbázis bejegyzéseinek összefésült másolatát a fájlban meghatározott módon és betölti az eredményül kapott adatokat a kernel biztonsági táblázataiba.

Tartományok közötti hozzárendelés:

Olyan környezet tervezésekor, amelyben az RBAC adatokat két tartomány biztosítja, mint például helyi fájlok és LDAP, figyelembe kell venni az entitások tartományok közötti hozzárendelésének problémáját. A tartományok közötti hozzárendelésre példa az LDAP címtárban meghatározott szerep hozzárendelése helyi felhasználóhoz vagy helyileg meghatározott szerep LDAP felhasználóhoz rendelése.

Távoli entitás (LDAP szerep) helyi entitáshoz rendelése nem jelent problémát, mivel nincs hatással a környezet más rendszereire. Azonban helyi entitás (helyi szerep) távoli entitáshoz (LDAP felhasználó) rendelése csak nagy körültekintéssel hajtható végre. Mivel a távoli entitás (LDAP felhasználó) több kliensen is látható, nincs garancia arra, hogy a hozzá rendelt helyi entitás (helyi szerep) meg van határozva vagy minden kliens rendszeren azonos a meghatározása. Egy szerep például helyileg meghatározható minden egyes kliensen, de a hozzájuk társított felhatalmazás eltérő lehet. A helyi szerephez rendelt távoli felhasználó emiatt eltérő felhatalmazásokkal rendelkezik minden kliensen, ennek pedig nem kívánatos biztonsági következményei lehetnek.

A helyi entitás LDAP entitáshoz rendeléséből fakadó lehetséges esetleges problémák megakadályozása érdekében ajánlatos, hogy az LDAP szerver valósítsa meg a hozzáférés-felügyeletet az RBAC adatbázisokhoz, így megakadályozva az egyes klienseknek a bejegyzések módosítását. Csak a privilegizált fiókon keresztül az LDAP szerverhez csatlakozó klienseknek szabad engedélyezni az LDAP RBAC entitásainak módosítását. Más klienseknek csak olvasási hozzáférésük lehet az LDAP RBAC adatbázisokhoz.

Méretkorlátok a kiterjesztett RBAC-ban:

A következő táblázat felsorolja az RBAC-hoz kapcsolódó elemek különféle korlátait:

10. táblázat: Az RBAC-hoz kapcsolódó elemek különböző korlátozásai

Leírás	Maximális méret
Szerepnév	63 nyomtatható karakter
Szerepek munkamenetenkénti maximális száma	8
Felhatalmazásnév maximális mérete	63 nyomtatható karakter
A felhatalmazáshierarchia szintjeinek maximális száma	9
A hozzáférési felhatalmazások parancsonkénti maximális száma	8
A felhatalmazott jogosultsághalmazok parancsonkénti maximális száma	8

Kiterjesztett RBAC adminisztrálása:

Ez a szakasz az RBAC adminisztrálásakor gyakori parancssor-használati szituációkat ír le. Ezek a példák a funkcionalitás főbb szempontjait szemléltetik. SMIT felületek szintén biztosítottak az RBAC adminisztrációhoz. Az RBAC SMIT menük gyorselérése: smit rbac.

Felhasználó által megadott felhatalmazások létrehozása:

A parancsok végrehajtásának vezérléséhez használható felhasználó által megadott felhatalmazásokat hozhat létre.

Az mkauth paranccsal felhasználó által megadott felhatalmazásokat hozhat létre. A felhatalmazási adatbázis módosításai azután lépnek életbe, hogy a módosításokat a **setkst** paranccsal letöltötte a kernelbe.

- Futtassa a következő parancsot felhasználó által megadott felhatalmazás létrehozásához:
  mkauth *felhatalmazásnév*

Szerepek létrehozása és módosítása:

Az **mkrole** parancs segítségével hozható létre szerep.

A szerepek az **mkrole** paranccsal kerülnek létrehozásra. A szerepadatbázis módosításai a **setkst** paranccsal a kernelbe történő letöltésük után lépnek életbe. A szerepeket a **chrole** paranccsal módosíthatja.

- Szerep létrehozásához futtassa a következő parancsot:
  mkrole dflt_msg="*Saját szerepem*" *szerepnév*
- Szerep létrehozásához és a felhatalmazás meglévő szerepből történő örököltetéséhez futtassa a következő parancsot:
  mkrole rolelist=*utódszerep1***,***utódszerep2 szerepnév*
- Szerepmeghatározás módosításához futtassa a következő parancsot:
  chrole rolelist=*utódszerep szerepnév*

Felhatalmazások szerepekhez rendelése:

Az **mkrole** vagy **chrole** parancs segítségével felhatalmazások rendelhetők egy szerephez.

- Futtassa az **mkrole** parancsot az **felhatalmazásnév1** és **felhatalmazásnév2** felhatalmazás **szerepnév** szerephez rendeléséhez:
  ```
  mkrole authorizations=felhatalmazásnév1,felhatalmazásnév2 szerepnév
  ```
- Futtassa az **chrole** parancsot a **felhatalmazásnév1** és **felhatalmazásnév2** felhatalmazás **szerepnév** szerephez rendeléséhez:
  ```
  chrole authorizations=felhatalmazásnév1,felhatalmazásnév2 szerepnév
  ```

Szerep hitelesítési módjának beállítása:

A szerepek aktiválása a szerep **auth_mode** attribútumával vezérelhető.

Az **auth_mode** attribútum érvényes értékei:

NONE Nincs szükség hitelesítésre

INVOKER
A hívóknak meg kell adniuk saját jelszavukat. Ez az alapértelmezés.

Adja ki a következő parancsot a felhasználók hitelesítésének kikényszerítéséhez adott szerep elfogadásakor:

chrole auth_mod=INVOKER *szerepnév*

Szerepek felhasználókhoz rendelése:

A **chuser** parancs segítségével szerepeket rendelhet felhasználóhoz.

A következő parancs futtatásával a **szerepnév1** és **szerepnév2** szerepek a **felhasználónév** nevű felhasználóhoz rendelhetők:

```
chuser
roles=szerep1,szerep2 felhasználónév
```

Szerepek aktiválása:

Alapértelmezésben a felhasználónak aktiválnia kell a szerepet a munkamenetben a privilegizált parancsok végrehajtása érdekében.

- A **szerepnév1** és **szerepnév2** szerepek aktiválásához futtassa a következő parancsot:
  ```
  swrole szerepnév1,szerepnév2
  ```
- A felhasználókhoz rendelt szerepek közül néhány alapértelmezett szerepnek van minősítve. Ezek a szerepek a felhasználó bejelentkezésekor automatikusan aktiválásra kerülnek. Ezek a szerepek a teljes bejelentkezési munkamenet során aktívak. Futtassa a következő parancsot a szerepnév1 felhasználóhoz rendeléséhez alapértelmezett szerepként:
  ```
  chuser roles=szerepnév1,szerepnév2
  default_roles=szerepnév1 felhasználónév
  ```

Az aktív szerephalmaz listázása:

A **rolelist** parancsot a **-e** kapcsolóval használva információkat jeleníthet meg egy munkamenet hatályos aktív szerephalmazáról.

- Egy munkamenet hatályos aktív szerephalmazának megjelenítéséhez futtassa a következő parancsot:
  ```
  rolelist -e
  ```

Felhasználó szerepeinek felsorolása:

A **rolelist** parancs szerep- és felhatalmazásinformációkat biztosít egy felhasználó aktuális szerepeiről vagy a hozzárendelt szerepekről.

Alapértelmezésben a **rolelist** parancs a felhasználóhoz rendelt szerepek listáját jeleníti meg. Ezek az információk alapvetően egyeznek az lsuser -a roles user1 parancs által megjelenítettekkel azzal a kivétellel, hogy tartalmazzák a szerep szöveges leírását, amennyiben az meg lett adva.

- A társított szerepek és felhatalmazások listájának megjelenítéséhez futtassa a következő parancsot:
  ```
  rolelist -a
  ```

Munkamenetszerepek megfigyelése:

A bejelentkezési munkameneteben aktív szerepek más attribútumokkal, például a UID és GID attribútummal együtt kerülnek megfigyelésre. Az **auditpr** parancs segítségével felsorolhatja ezeket a szerepeket.

A szerepek nyomkövetési naplóból való megjelenítéséhez futtassa a következő parancsot:

```
auditpr -h eli -i /audit/trail
```

Felhatalmazások futó folyamathoz rendelése:

A **setsecattr** parancs segítségével módosíthatja egy futó folyamat jogosultságait.

- Folyamathoz tartozó hatályos jogosultsághalmazok frissítéséhez futtassa a következő parancsot:

  setsecattr –p eprivs=*jogosultságok pid*
- Egy jogosultság folyamat hatályos jogosultsághalmazához adása előtt biztosítani kell, hogy a jogosultság már létezzen a maximális jogosultsághalmazban. A maximális jogosultsághalmaz módosításához futtassa az alábbi parancsot:

  setsecattr –p mprivs=*jogosultságok pid*

WPAR jogosultságok adminisztrálása:

Minden WPAR-hoz jogosultságok halmaza tartozik, amely meghatározza a hatáskört. Ezt WPAR jogosultsághalmaznak (WPS) nevezik.

Az adott WPAR-on belül futó folyamatok csak a WPS-ben elérhető jogosultságokat használhatják.

- A WPS globális WPAR-ból való módosításához futtassa a következő parancsot:

  chwpar –S privs+=*jogosultságok wpar_név*

Parancs által igényelt jogosultságok meghatározása:

Egyes parancsok speciális jogosultságokat igényelnek privilegizált műveletek végrehajtásához. A jogosultságok segítségével a kernel átlépi a biztonsági korlátozásokat.

A **tracepriv** parancs segítségével meghatározhatja egy parancs sikeres futtatásához szükséges jogosultságokat. A **tracepriv** parancs rögzíti a futtatásakor a másik parancs által használt jogosultságokat. A parancsot **PV_ROOT** jogosultsággal kell futtatni, így a jogosultságok használatára tett kísérletek sikeresek lesznek. A parancs befejeződésekor a használt jogosultságok halmaza elküldésre kerül az stdout kimenetre.

- Adott parancs profilozásához futtassa a következő parancsot:

  tracepriv –ef *parancsnév*

Felhatalmazás használata a parancsok vezérléséhez:

A felhatalmazás segítségével vezérelhető a parancsok futása.

A **setsecattr** parancs segítségével felhatalmazások társíthatók a parancshoz. A **setsecattr** parancs egy szakaszt ad a privilegizált parancsadatbázishoz (/etc/security/privcmds). A **setkst** paranccsal az adatbázis módosításait le kell tölteni a kernelre.

- Felhatalmazások parancshoz adásához futtassa a következő parancsot:

  setsecattr –c accessauths=*felhatalmazásnevek* **innateprivs=***jogosultságok* **proxyprivs=***jogosultságok* **authprivs=***felhatalmazásnév***=***jogosultságok parancsnév*

Eszközök elérésének vezérlése:

Az RBAC lehetőséget ad az eszközök elérésének további felügyeletére. A rendszeradminisztrátor megadhatja az eszköz olvasási vagy írási módban történő megnyitásához szükséges jogosultságokat.

A DVD író írási elérése például a **PV_DEV_CONFIG** jogosultsággal vezérelhető, így csak az ezzel a jogosultsággal rendelkező folyamatok hozhatnak létre DVD-ket.

- Eszköz eszközadatbázishoz adásához futtassa a következő parancsot:

  setsecattr –d readprivs=*jogosultságok* **writeprivs=***jogosultságok eszköznév*

RBAC Kernel biztonsági táblázatok frissítése:

A **setkst** parancs beolvassa a biztonsági adatbázist és betölti az adatbázis információit a Kernel biztonsági táblákba (KST).

Alapértelmezésben minden biztonsági adatbázis átküldésre kerül a KST-be. A **-t** paraméterrel adott adatbázis is megadható. Annak megadása, hogy csak a felhatalmazási adatbázis kerüljön átküldésre a KST-be, szintén frissíti a szerep és privilegizált parancsadatbázisokat a KST-ben, mivel a szerep és a privilegizált parancsadatbázis a felhatalmazási adatbázistól függ.

- Az összes legújabb RBAC adatbázis kernelbe küldéséhez futtassa a következő parancsot:

  ```
  setkst
  ```

A kiterjesztett RBAC mód kapcsoló használata:

Egy rendszerszintű konfigurációs kapcsoló segítségével letilthatók a kiterjesztett RBAC képességek és vissza lehet térni az örökölt RBAC viselkedéshez.

A rendszeradminisztrátor letilthatja a kiterjesztett RBAC módot a **chdev** parancs futtatásával a **sys0** eszközön és az **enhanced_RBAC** attribútumot **false** értékkel megadva, majd a rendszert újraindításával. A mód visszaváltható a kiterjesztett RBAC módra az **enhanced_RBAC** attribútum **true** értékre állításával és a rendszer újraindításával.

- Futtassa a következő parancsot a visszatéréshez az örökölt RBAC módhoz:

  ```
  chdev -l sys0 -a enhanced_RBAC=false
  ```

- Az **enhanced_RBAC** attribútum értékének kiírásához futtassa a következő parancsot:

  ```
  lsattr -E -l sys0 -a enhanced_RBAC
  ```

WPAR környezetben az RBAC mód csak a globális rendszerből állítható be és a globális módot, valamint az összes WPAR-t érinti.

Megjegyzés: A kiterjesztett RBAC mód letiltása csökkentheti a rendszer biztonsági küszöbét, különösen WPAR-on belül.

RBAC-vel kapcsolatos parancsok

Az alábbi táblázat felsorolja azokat az RBAC szolgáltatással kapcsolatos parancsokat, amelyek rendelkezésre állnak az AIX operációs rendszerben az RBAC keretrendszer kezeléséhez és használatához.

Parancs	Leírás
chauth	felhasználó által megadott felhatalmazási attribútumok módosítása
chrole	Szerepattribútumok módosítása
ckauth	Az aktuális folyamat ellenőrzése a felhatalmazásra vonatkozóan
lsauth	Felhasználó és rendszer által meghatározott felhatalmazási attribútumok megjelenítése
lskst	A Kernel biztonsági táblázatokban lévő bejegyzések felsorolása
lspriv	A rendszeren rendelkezésre álló jogosultságok megjelenítése
lsrole	Szerepattribútumok megjelenítése
lssecattr	Parancs, eszköz, folyamat vagy fájl biztonsági attribútumainak megjelenítése

Parancs	Leírás
mkauth	Új felhasználó által megadott felhatalmazás létrehozása
mkrole	Új szerep létrehozása
pvi	Privilegizált fájlszerkesztő
rbacqry	RBAC engedélyezése alkalmazásokhoz
rbactoldif	RBAC felhasználószintű adatbázisok kiadása LDAP-nak megfelelő formátumban
rmauth	Felhasználó által megadott felhatalmazások eltávolítása
rmrole	Szerep eltávolítása
rmsecattr	Parancs, eszköz vagy fájl biztonsági attribútumaihoz tartozó meghatározás eltávolítása
rolelist	Felhasználó vagy folyamat szerepinformációinak megjelenítése
setkst	Az RBAC felhasználószintű adatbázisokban lévő bejegyzések Kernel biztonsági táblázatokba küldése
setsecattr	Parancs, eszköz, folyamat vagy fájl biztonsági attribútumainak beállítása
setsecconf	Kernelbiztonsági kapcsolók módosítása
swrole	Új szerepmunkamenet létrehozása
tracepriv	A parancs sikeres futtatásához szükséges jogosultságok nyomkövetése

RBAC-hez kapcsolódó fájlok

A következő táblázat az AIX rendszeren az adatbázis-információk beállításához és tárolásához biztosított RBAC-vel kapcsolatos fájlokat sorolja fel.

Fájl	Leírás
/etc/nscontrol.conf	Névszolgáltatás-vezérlő fájl adott biztonsági adatbázisokhoz
/etc/security/authorizations	Felhasználó által megadott felhatalmazási adatbázis
/etc/security/privcmds	Privilegizált parancsadatbázis
/etc/security/privfiles	Privilegizált fájladatbázis
/etc/security/privdevs	Privilegizált eszközadatbázis
/etc/security/roles	Szerepadatbázis

Kiterjesztett RBAC használata alkalmazásokban

Számos alkalmazás nem igényel módosítást a kiterjesztett RBAC környezetben való sikeres futáshoz. Elegendő lehet, hogy egyszerűen megadja az alkalmazás hozzáférési felhatalmazásait és a hozzátartozó jogosultságokat, majd hozzárendeli az alkalmazást a privilegizált parancsadatbázishoz.

Az alkalmazás használhatja a kiterjesztett RBAC-t az RBAC felületek meghívásával az alkalmazás végrehajtásának finomabb szintű vezérléséhez, ezáltal biztonságosabb alkalmazás érhető el. Alkalmazások, amelyek számára a kiterjesztett RBAC-vel való integráció előnyös lehet:

- Alkalmazások, amelyek használata a root felhasználóra vagy egy adott csoport tagjaira van korlátozva. Ezek az alkalmazások jellemzően hatályos felhasználóazonosságot vagy csoporttagságot keresnek, és módosíthatók, hogy inkább felhatalmazást keressenek.
- Alkalmazások, amelyek kihasználják a **setuid** vagy **setgid** módbitet annak lehetővé tétele érdekében, hogy a jogosulatlan felhasználók jogosultságot kapjanak a parancs meghívása során. Ezek az alkalmazások jogosultságbefogással biztonságosabbá tehetők, így kevesebb jogosultság kerül felhasználásra a feladat végrehajtásához.

Felhatalmazásellenőrzés:

A privilegizált műveletek végrehajtási lehetőségét a hívó felhasználó felhasználói azonosítójának vagy csoportazonosítójának használatával meghatározó alkalmazásokat módosítani kell, hogy azok a felhatalmazást ellenőrizzék.

Tételezzünk fel egy alkalmazást, amely fájlrendszer-konfigurációs feladatokat végez és pillanatnyilag a root felhasználónak (UID = 0) engedélyezi privilegizált műveletek végrehajtását.

```
if (getuid() == 0) {
   /* privilegizált műveletek folytatásának engedélyezése */
}
```

Ahhoz, hogy ez az alkalmazás ehelyett adott felhatalmazással (**aix.fs.config**) rendelkező felhasználóknak tegye lehetővé a privilegizált műveletet, a **checkauths** alkalmazás programozási felületet kell használni a felhatalmazásellenőrzés végrehajtására:

```
if (checkauths("aix.fs.config", CHECK_ALL)) {
  /* privilegizált műveletek folytatásának engedélyezése */
 }
```

A **checkauths** alkalmazás programozási felület támogatja mind az örökölt, mind a kiterjesztett RBAC módokat és **0** kódot ad vissza, ha a hívó folyamat rendelkezik a megadott felhatalmazással. A **checkauths** alkalmazás programozási felület meghatározza, hogy a root felhasználó hatásköre engedélyezett-e, majd ennek megfelelően engedélyezi vagy tiltja a root felhasználónak a felhatalmazásellenőrzések kihagyását. Az AIX Version 6.1 előtt a **MatchAllAuths**, **MatchAnyAuths**, **MatchAllAuthsList** és **MatchAnyAuthsList** alkalmazás programozási felületek voltak használhatók a felhatalmazásellenőrzések végrehajtására. Az AIX Version 6.1 és újabb változatokhoz készített alkalmazásoknak a **checkauths** alkalmazás programozási felületet kell használniuk, mivel az támogatja az örökölt és kiterjesztett RBAC módokat, valamint a root letiltását.

A fenti példához hasonlóan az adott feladat végrehajtását bizonyos felhasználóknak a **getuid**, **getgid** vagy hasonló függvények hívásával biztosító alkalmazásokat is módosítani kell, hogy azok a **checkauths** alkalmazás programozási felület segítségével felhatalmazásellenőrzést hajtsanak végre. Ha az ellenőrzendő felhasználói azonosító vagy csoportazonosító nem a root felhasználóé, akkor a **sys_parm** rendszerhívással lekérdezhető, hogy a kiterjesztett RBAC engedélyezett-e. Ha a kiterjesztett RBAC nincs engedélyezve, akkor a kód végrehajthatja a már meglévő ellenőrzéseket. Ellenkező esetben, ha a kiterjesztett RBAC engedélyezve van, akkor a kód ellenőrizheti az érintett rendszer- vagy felhasználó által megadott felhatalmazásokat.

Jogosultságbefogás:

Az alkalmazások módosítás után a felhatalmazások ellenőrzése érdekében tovább módosíthatók a részletes jogosultságbefogás működés közbeni használatához.

Az alkalmazások a **priv_raise** alkalmazás programozási felület segítségével növelik a művelet végrehajtásához szükséges jogosultságokat és a **priv_lower** alkalmazás programozási felület segítségével csökkentik azokat. A jogosultságok privilegizált művelet végrehajtására tett kísérlet előtti azonnali növelése, illetve a művelet befejeződése utáni csökkentése jogosultságbefogás néven ismert, és az alkalmazások jogosultsághasználatának előnyben részesített módszere. Jogosultság növeléséhez a jogosultságnak elérhetőnek kell lennie az alkalmazás maximális jogosultsághalmazában, a privilegizált parancsadatbázisban. A jogosultság növelése a jogosultságot a folyamat hatályos jogosultsághalmazában (EPS) helyezi el. A jogosultság csökkentése eltávolítja azt az EPS-ből. A következő kódminta a jogosultságbefogást mutatja be az **auditproc** API körül.

```
priv_raise(PV_AU_ADMIN, -1); /*
jogosultság növelése szükség esetén*/
auditproc();     /* megfigyelési rendszerhívás meghívása */
priv_lower(PV_AU_ADMIN, -1); /* jogosultság csökkentése */
```

RBAC használatára képes alkalmazások:

Hagyományosan AIX és más root felhasználóval rendelkező kiterjesztett RBAC rendszereken a privilegizált parancsadatbázisból hiányzó, a root vagy root által birtokolt **setuid** programok (amely felhasználói azonosítója 0) mindig megkapják az összes jogosultságot a kernelben. A kernel jogosultságellenőrzései emiatt mindig sikeresen térnek vissza, akkor is, ha a kért jogosultság nincs jelen a folyamat hatályos jogosultsághalmazában (EPS).

Ez a viselkedés továbbra is szükséges a **setuid** alkalmazások támogatásához, de biztonsági kockázatot jelenthet, mivel a **setuid** programok a root minden hatáskörével rendelkeznek.

Megfelelő jogosultságbefogás lehetővé tételéhez egy root felhasználóval rendelkező kiterjesztett RBAC rendszeren a folyamatstruktúrában egy új bit került bevezetésre. Ha ez a bit be van állítva, akkor a folyamat RBAC használatára képes folyamattá válik, a 0 hatályos felhasználói azonosító pedig nem biztosít többletjogosultságokat. Ez a bit a programban a **proc_rbac_op** rendszerhívás használatával állítható be. A privilegizált parancsadatbázisban nem szereplő **setuid** programok ezen funkcionalitás használatával csökkenthetik a biztonsági sebezhetőséget az elérhető jogosultságok csökkentésével. Ne feledje el, hogy a privilegizált parancsadatbázisban meghatározott programok automatikusan RBAC használatára képesként kerülnek megjelölésre és csak az adatbázisban felsorolt jogosultságokat kapják meg.

A következő kód bemutatja, hogy egy alkalmazás hogyan jelölheti meg magát RBAC használatára képesként, majd hogyan hajthatja végre a megfelelő jogosultságbefogást:

```
#include <userpriv.h
#include <sys/priv.h>

privg_t effpriv;

int rbac_flags = SEC_RBACAWARE;

/* A folyamat megjelölése RBAC használatára képesként. */
proc_rbac_op(-1, PROC_RBAC_SET, &rbac_flags);

/* A hatályos jogosultság beállítása üresként. */
priv_clrall(effpriv);
setppriv(-1, &effpriv, NULL, NULL, NULL);

/* Jogosultság emelése szükség esetén. */
priv_raise(PV_AU_ADMIN, -1);
auditproc();

/* Jogosultság csökkentése, ha már nem szükséges. */
priv_lower(PV_AU_ADMIN, -1);
```

RBAC API-k:

A rendszeren az RBAC-hez kapcsolatódó elérhető API-kat az alábbi táblázat tartalmazza. További információkért tekintse meg az adott API-kat.

API	Leírás
checkauths	A felhatalmazások átadott listáját összehasonlítja az aktuális folyamathoz tartozó felhatalmazások.
GetUserAuths	Lekéri az aktuális folyamathoz rendelt felhatalmazások halmazát.
MatchAllAuths, MatchAllAuthsList, MatchAnyAuths, MatchAnyAuthsList	Összehasonlítja a felhatalmazásokat. Ezen alkalmazás programozási felületekhez a checkauths API az előnyben részesített.
getauthattr, putauthattr	Lekérdezi vagy módosítja a felhatalmazási adatbázisban meghatározott felhatalmazásokat.

API	Leírás
getauthattrs	Felhatalmazási attribútumokat kér le a felhatalmazási adatbázisból.
putauthattrs	Frissíti a felhatalmazási attribútumokat a felhatalmazási adatbázisban.
getcmdattr, putcmdattr	Lekérdezi vagy módosítja a parancs biztonsági információit a privilegizált parancsadatbázisban.
getcmdattrs	Parancsattribútumokat kér le a privilegizált parancsadatbázisból.
putcmdattrs	Frissíti a parancsattribútumokat a privilegizált parancsadatbázisban.
getdevattr, putdevattr	Lekérdezi vagy módosítja az eszköz biztonsági információit a privilegizált eszközadatbázisban.
getdevattrs	Eszközattribútumokat kér le a privilegizált eszközadatbázisból.
putdevattrs	Frissíti az eszközattribútumokat a privilegizált eszközadatbázisban.
getpfileattr, putpfileattr	Lekérdezi vagy módosítja a fájl biztonsági információit a privilegizált fájladatbázisban.
getpfileattrs	Fájlattribútumokat kér le a privilegizált fájladatbázisból.
putpfileattrs	Frissíti a fájlattribútumokat a privilegizált fájladatbázisban.
getroleattr, putroleattr	Lekérdezi vagy módosítja a szerepadatbázisban meghatározott szerepeket.
getroleattrs	Szerepattribútumokat kér le a szerepadatbázisból.
putroleattrs	Frissíti a szerepattribútumokat a szerepadatbázisban.
getsecorder	Lekéri a tartományok sorrendjét bizonyos biztonsági adatbázisokhoz.
setsecorder	Beállítja a tartományok sorrendjét adott biztonsági adatbázisokhoz.

AIX jogosultságok

Az AIX alatt elérhető jogosultságokat az alábbi táblázat sorolja fel. Az egyes jogosultságok leírása és a kapcsolódó rendszerhívások biztosítottak. Egyes jogosultságok hierarchiát alkotnak, amelyben egy jogosultság megadhatja a másik jogosultsághoz tartozó összes jogot.

A jogosultságok ellenőrzésekor a rendszer először meghatározza, hogy a folyamat rendelkezik-e a legalacsonyabb szükséges jogosultsággal, majd folytatja a hierarchia ellenőrzését magasabb szintű jogosultságokat keresve. A **PV_AU_** jogosultságú folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultságú folyamat automatikusan rendelkezik az összes felsorolt jogosultsággal a **PV_SU_** jogosultság kivételével.

Jogosultság	Leírás	Rendszerhívás-hivatkozás
PV_ROOT	A folyamatnak megadja az alább felsorolt összes jogosultságot a PV_SU_ (és az általa meghatározott jogosultságok) kivételével	
PV_AU_ADD	Engedélyezi a folyamatnak a megfigyelési rekordok rögzítését/felvételét	auditlog
PV_AU_ADMIN	Engedélyezi a folyamatnak a megfigyelési rendszer konfigurálását és lekérdezését	audit, auditbin, auditevents, auditobj

Jogosultság	Leírás	Rendszerhívás-hivatkozás
PV_AU_PROC	Engedélyezi a folyamatnak egy folyamat megfigyelési állapotának beállítását vagy lekérdezését	auditproc
PV_AU_READ	Engedélyezi a folyamatnak a Trusted AIX alatt megfigyelési fájlként jelölt fájl olvasását	
PV_AU_WRITE	Engedélyezi a folyamatnak a Trusted AIX alatt megfigyelési fájlként jelölt fájl írását vagy törlését	
PV_AU_	Egyenértékű a fenti megfigyelési jogosultságok (PV_AU_*) kombinációjával	
PV_AZ_ADMIN	Engedélyezi a folyamatnak a kernel biztonsági táblák módosítását	sec_setkst
PV_AZ_READ	Engedélyezi a folyamatnak a kernel biztonsági táblák lekérését	sec_getkat, sec_getkpct, sec_getkpdt, sec_getkrt, stb.
PV_AZ_ROOT	Hatására a folyamat átmegy a jogosultságellenőrzéseken az exec() során (öröklési célokra használatos)	
PV_AZ_CHECK	Hatására a folyamat átmegy az összes jogosultságellenőrzésen	sec_checkauth
PV_DAC_R	Engedélyezi a folyamatnak a DAC olvasással kapcsolatos korlátozások felülbírálását	access, creat, accessx, open, read, faccessx, mkdir, getea, rename, statx, _sched_getparam, _sched_getscheduler, statea, listea
PV_DAC_W	Engedélyezi a folyamatnak a DAC írással kapcsolatos korlátozások felülbírálását	A fentiek közül sok, valamint a következők: setea, write, symlink, _setpri, _sched_setparam, _sched_setscheduler, fsetea, rmdir, removeea
PV_DAC_X	Engedélyezi a folyamatnak a DAC végrehajtási korlátozások felülbírálását	A fentiek közül sok, valamint: execve, symlink, rmdir, chdir, fchdir, ra_execve
PV_DAC_O	Engedélyezi a folyamatnak a DAC tulajdonjog-korlátozások felülbírálását	chmod, utimes, setacl, revoke, mprotect
PV_DAC_UID	Engedélyezi a folyamat felhasználói azonosítójának módosítását	setuid, seteuid, setuidx, setreuid, ptrace64
PV_DAC_GID	Engedélyezi a folyamatnak új csoportazonosító beállítását vagy a meglévő módosítását	setgid, setgidx, setgroups, ptrace64
PV_DAC_RID	Engedélyezi a folyamatnak új szerepazonosító beállítását vagy a meglévő módosítását	setroles, getroles
PV_DAC_	Egyenértékű a fenti DAC jogosultságok (PV_DAC_*) kombinációjával	
PV_FS_MOUNT	Engedélyezi a folyamatnak fájlrendszer felépítését és lebontását	vmount, umount
PV_FS_MKNOD	Engedélyezi a folyamatnak tetszőleges típusú fájl létrehozását vagy az mknod rendszerhívás végrehajtását	mknod
PV_FS_CHOWN	Engedélyezi a folyamatnak egy fájl tulajdonjogának módosítását	chown, chownx, fchownx, lchown

Jogosultság	Leírás	Rendszerhívás-hivatkozás
PV_FS_QUOTA	Engedélyezi a folyamatnak lemezkvótákhoz kapcsolódó műveletek kezelését	quotactl
PV_FS_LINKDIR	Engedélyezi a folyamatnak könyvtárra mutató közvetlen hivatkozás létrehozását	link, unlink, remove
PV_FS_CNTL	Engedélyezi a folyamatnak különböző vezérlőműveletek végrehajtását a fájlrendszeren a kiterjesztés és összehúzás kivételével	fscntl
PV_FS_RESIZE	Engedélyezi a folyamatnak kiterjesztés és összehúzás típusú műveletek végrehajtását a fájlrendszeren	fscntl
PV_FS_CHROOT	Engedélyezi a folyamatnak a gyökérkönyvtárának módosítását	chroot
PV_FS_PDMODE	Engedélyezi a folyamatnak particionált típusú könyvtár létrehozását vagy beállítását	pdmkdir
PV_FS_	Egyenértékű a fenti fájlrendszer-jogosultságok (PV_FS_*) kombinációjával	
PV_PROC_PRIV	Engedélyezi a folyamatnak egy folyamathoz tartozó jogosultsághalmazok módosítását vagy megjelenítését	setppriv, getppriv
PV_PROC_PRIO	Engedélyezi a folyamatnak vagy szálnak a prioritás, irányelv és más ütemezési paraméterek módosítását	_prio_requeue, _setpri, _setpriority, _getpri, _sched_setparam, _sched_setscheduler, _thread_setsched, thread_boostceiling, thread_setmystate, thread_setstate
PV_PROC_CORE	Engedélyezi a folyamatnak a tárkiíratást	gencore
PV_PROC_RAC	Engedélyezi a folyamatnak a felhasználónkénti korlátnál több folyamat létrehozását	appsetrlimit, setrlimit64, mlock, mlockall, munlock, munlockall, plock, upfget, upfput, restart, brk, sbrk
PV_PROC_RSET	Engedélyezi a folyamatnak erőforráshalmaz (rset) csatlakoztatását folyamathoz vagy szálhoz	bindprocessor, ra_attachrset, ra_detachrset, rs_registername, rs_setnameattr, rs_discardname, rs_setpartition, rs_getassociativity, kra_mmapv
PV_PROC_ENV	Engedélyezi a folyamatnak felhasználói információk beállítását a felhasználói adatszerkezetben	ue_proc_register, ue_proc_unregister, usrinfo
PV_PROC_CKPT	Engedélyezi a folyamatnak ellenőrzési pont készítését másik folyamatról vagy annak újraindítását	setcrid, restart
PV_PROC_CRED	Engedélyezi a folyamatnak hitelesítési attribútumok beállítását	__pag_setvalue, __pag_setvalue64, __pag_genpagvalue
PV_PROC_SIG	Engedélyezi a folyamat számára jelzés küldését egy nem kapcsolódó folyamatnak	_sigqueue, kill, signohup, gencore, thread_post, thread_post_many
PV_PROC_TIMER	Engedélyezi a folyamatnak finomabb beállítású időmérők elküldését és használatát	appresabs, appresinc, absinterval, incinterval, _poll, _select _timer_settime

Jogosultság	Leírás	Rendszerhívás-hivatkozás
PV_PROC_RTCLK	Engedélyezi a folyamatnak a CPU-idő óra elérését	_clock_getres, _clock_gettime, _clock_settime, _clock_getcpuclockid
PV_PROC_VARS	Engedélyezi a folyamatnak a folyamat által hangolható paraméterek lekérését és frissítését	smttune
PV_PROC_PDMODE	Engedélyezi a folyamatnak a particionált könyvtár VALÓS módjának módosítását	setppdmode
PV_PROC_	Egyenértékű a fenti folyamatjogosultságok (PV_PROC_*) kombinációjával	
PV_TCB	Engedélyezi a folyamatnak a kernel megbízható könyvtárútvonalának módosítását	chpriv, fchpriv
PV_TP	Jelzi, hogy a folyamat megbízható útvonalú folyamat és lehetővé teszi a megbízható útvonalú folyamatokra korlátozott műveleteket. (Megjegyzés: ugyanaz, mint a régi AIX BYPASS_TPATH jogosultság)	
PV_WPAR_CKPT	Engedélyezi a folyamatnak az ellenőrzőpont/újraindítás művelet végrehajtását a WPAR-ban	smcr_proc_info, smcr_exec_info, smcr_mapinfo, smcr_net_oper, smcr_procattr, aio_suspend_io, aio_resume_io
PV_KER_ACCT	Engedélyezi a folyamatnak a számlázó alrendszerre vonatkozó korlátozott műveletek végrehajtását	acct, _acctctl, projctl
PV_KER_DR	Engedélyezi a folyamatnak a dinamikus újrakonfigurálási műveletek meghívását	_dr_register, _dr_notify, _dr_unregister, dr_reconfig
PV_KER_TIME	Engedélyezi a folyamatnak a rendszeróra és rendszeridő módosítását	adjtime, appsettimer, _clock_settime
PV_KER_RAC	Engedélyezi a folyamatnak nagy (nem lapozható) oldalak használatát a megosztott memóriaszegmensekhez	shmctl, vmgetinfo
PV_KER_WLM	Engedélyezi a folyamatnak a WLM konfiguráció inicializálását és módosítását	_wlm_set, _wlm_tune, _wlm_assign
PV_KER_EWLM	Engedélyezi a folyamatnak az eWLM környezet inicializálását vagy lekérdezését	
PV_KER_VARS	Engedélyezi a folyamatnak a kernel futás közben hangolható paramétereinek vizsgálatát vagy beállítását	sys_parm, getkerninfo, __pag_setname, sysconfig, kunload64
PV_KER_REBOOT	Engedélyezi a folyamatnak a rendszer leállítását	reboot
PV_KER_RAS	Engedélyezi a folyamatnak a RAS rekordok konfigurálását vagy írását, hibanaplózást, nyomkövetést és a kiíratási funkciókat	mtrace_set, mtrace_ctl
PV_KER_LVM	Engedélyezi a folyamatnak az LVM alrendszer beállítását	
PV_KER_NFS	Engedélyezi a folyamatnak az NFS alrendszer beállítását	

Jogosultság	Leírás	Rendszerhívás-hivatkozás
PV_KER_VMM	Engedélyezi a folyamatnak a lapozási paraméterek és a VMM által hangolható paraméterek módosítását a kernelben	swapoff, _swapon_ext, vmgetinfo
PV_KER_WPAR	Engedélyezi a folyamatnak munkaterület-partíció beállítását	brand, corral_config, corral_delete, corral_modify, wpar_mkdevexport, wpar_rmdevexport, wpar_lsdevexport
PV_KER_CONF	Engedélyezi a folyamatnak különféle rendszerkonfigurációs műveletek végrehajtását	sethostname, sethostid, unameu, setdomainname
PV_KER_EXTCONF	Engedélyezi a folyamatnak különféle konfigurációs feladatok végrehajtását a kernel kiterjesztéseiben (a kernelkiterjesztési szolgáltatásokhoz)	
PV_KER_IPC	Engedélyezi a folyamatnak az IPC üzenetsor pufferértékének növelését és lehetővé teszi az shmget-nek a tartományokkal történő csatlakoztatást	msgctl, shm_open, shmget, ra_shmget, ra_shmgetv, shmctl
PV_KER_IPC_R	Engedélyezi a folyamatnak egy IPC üzenetsor, szemaforhalmaz vagy megosztott memóriaszegmens olvasását	msgctl, __msgrcv, _mq_open, semctl, shmat, shm_open, __semop, shmctl, __semtimedop, sem_post, _sem_wait, __msgrcv, __msgxrcv
PV_KER_IPC_W	Engedélyezi a folyamatnak egy IPC üzenetsor, szemaforhalmaz vagy megosztott memóriaszegmens írását	_mq_open, shmat, _sem_open, semctl, shm_open, shmctl, mq_unlink, sem_unlink, shm_unlink, msgctl, __msgsnd
PV_KER_IPC_O	Engedélyezi a folyamatnak a DAC tulajdonjog felülbírálását az összes IPC objektumon	msgctl, semctl, shmctl, fchmod, fchown
PV_KER_SECCONFIG	Engedélyezi a folyamatnak a kernel biztonsági kapcsolóinak beállítását	sec_setsecconf, sec_setrunmode, sec_setsyslab, sec_getsyslab
PV_KER_PATCH	Engedélyezi a folyamatnak a kernelbővítmények javítását	
PV_KER_	Egyenértékű a fenti kerneljogosultságok (PV_KER_*) kombinációjával	
PV_DEV_CONFIG	Engedélyezi a folyamatnak a kernelbővítmények és -eszközök konfigurálását a rendszeren	sysconfig
PV_DEV_LOAD	Engedélyezi a folyamatnak a kernelkiterjesztések és -eszközök betöltését és eltávolítását a rendszeren	sysconfig
PV_DEV_QUERY	Engedélyezi a folyamatnak a kernelmodulok lekérdezését	sysconfig
PV_SU_ROOT	Megadja a folyamatnak a szabványos AIX felettes felhasználó összes jogosultságát	
PV_SU_EMUL	Megadja a folyamatnak a szabványos AIX felettes felhasználó összes jogosultságát, ha az UID 0	
PV_SU_UID	Hatására a getuid rendszerhívás 0 értéket ad vissza	getuidx

Jogosultság	Leírás	Rendszerhívás-hivatkozás
PV_SU_	Egyenértékű a felettes felhasználó összes fenti jogosultságának (PV_SU_*) kombinációjával	
PV_NET_CNTL	Engedélyezi a folyamatnak a hálózati táblázatok módosítását	socket, bind, listen, _naccept, econnect, ioctl, rmsock, setsockopt
PV_NET_PORT	Engedélyezi a folyamatnak a csatlakozást a privilegizált portokon	bind
PV_NET_RAWSOCK	Engedélyezi a folyamatnak a hálózati réteg közvetlen elérését	socket, _send, _sendto, sendmsg, _nsendmsg
PV_NET_CONFIG	Engedélyezi a folyamatnak a hálózati paraméterek beállítását	
PV_NET_	Egyenértékű a fenti hálózatkezelési jogosultságok (PV_NET_*) kombinációjával	

következő táblázatban felsorolt jogosultságok a Trusted AIX rendszerre jellemzők:

Trusted AIX jogosultság	Leírás	Rendszerhívás-hivatkozás
PV_LAB_CL	Engedélyezi a folyamatnak az alany SCL-ek módosítását, a folyamat engedélyének függvényében	
PV_LAB_CLTL	Engedélyezi a folyamatnak az alany TCL-ek módosítását, a folyamat engedélyének függvényében	
PV_LAB_LEF	Engedélyezi a folyamatnak a címke kódolásfájljának olvasását	
PV_LAB_SLDG	Engedélyezi a folyamatnak az SL-ek lefokozását a folyamat engedélyének függvényében	
PV_LAB_SLDG_STR	Engedélyezi a folyamatnak egy csomag SL-jének visszaléptetését a folyamat engedélyének függvényében	
PV_LAB_SL_FILE	Engedélyezi a folyamatnak az objektum SL-ek módosítását a folyamat engedélyének függvényében	
PV_LAB_SL_PROC	Engedélyezi a folyamatnak az alany SL-ek módosítását, a folyamat engedélyének függvényében	
PV_LAB_SL_SELF	Engedélyezi a folyamatnak a saját SL-jének módosítását, a folyamat engedélyének függvényében	
PV_LAB_SLUG	Engedélyezi a folyamatnak az SL-ek kiemelését a folyamat engedélyének függvényében	
PV_LAB_SLUG_STR	Engedélyezi a folyamatnak egy csomag SL-jének kiemelését a folyamat engedélyének függvényében	
PV_LAB_TL	Engedélyez a folyamatnak az alany és objektum TL-jének módosítását	

Trusted AIX jogosultság	Leírás	Rendszerhívás-hivatkozás
PV_LAB_	Egyenértékű a fenti címkejogosultságok (PV_LAB_*) kombinációjával	
PV_MAC_CL	Engedélyezi a folyamatnak az érzékenységengedély-korlátozások kihagyását	
PV_MAC_R_PROC	Engedélyezi a folyamatnak a MAC olvasási megszorítások kihagyását folyamatinformációk lekérésekor, ha a célfolyamat címkéje a működő folyamat engedélyén belül van	
PV_MAC_W_PROC	Engedélyezi a folyamatnak a MAC írási megszorítások kihagyását jelzés küldésekor a folyamatnak, ha a célfolyamat címkéje a működő folyamat engedélyén belül van	
PV_MAC_R	Engedélyezi a folyamatnak a MAC olvasási korlátozások kihagyását	
PV_MAC_R_CL	Engedélyezi a folyamatnak a MAC olvasási megszorítások kihagyását, ha az objektum címkéje a folyamat engedélyén belül van	
PV_MAC_R_STR	Engedélyezi a folyamatnak a MAC olvasási megszorítások kihagyását üzenet adatfolyamból való olvasásakor, ha az üzenet címkéje a folyamat engedélyén belül van	
PV_MAC_W	Engedélyezi a folyamatnak a MAC írási korlátozások kihagyását	
PV_MAC_W_CL	Engedélyezi a folyamatnak a MAC írási megszorítások kihagyását, ha az objektum címkéje a folyamat engedélyén belül van	
PV_MAC_W_DN	Engedélyezi a folyamatnak a MAC írási megszorítások kihagyását, ha a folyamat címkéje meghatározza az objektum címkéjét és az objektum címkéje a folyamat engedélyén belül van	
PV_MAC_W_UP	Engedélyezi a folyamatnak a MAC írási megszorítások kihagyását, ha a folyamat címkéjét meghatározza az objektum címkéje és az objektum címkéje a folyamat engedélyén belül van	
PV_MAC_OVRRD	Kihagyja a MAC korlátozásokat MAC alól kivételként jelzett fájlok esetén	
PV_MAC_	Egyenértékű a fenti MAC jogosultságok (PV_MAC_*) kombinációjával	
PV_MIC	Engedélyezi a folyamatnak az integritási korlátozások kihagyását	
PV_MIC_CL	Engedélyezi a folyamatnak az integritási engedélykorlátozások kihagyását	

Tartomány RBAC

Az AIX 6.1 változatában bevezetett Szerep alapú hozzáférés-felügyelet (RBAC) módot biztosít a root kiemelt felhasználó különféle funkcióinak felosztására szerepekbe, amelyek a rendszer más felhasználóinak delegálhatók. Az RBAC segítségével delegálhatók a feladatok és ezzel javul a rendszer biztonsága, mert könnyebb megfigyelni és követni a rendszeren zajló tevékenységeket. Az RBAC használatával átruházható a felelősség egy másik felhasználóra (a felhatalmazott felhasználóra), de nem korlátozhatók a felhatalmazott felhasználó adminisztrátori jogai a rendszer bizonyos erőforrásaira. Például a hálózatadminisztrátori jogokkal rendelkező felhasználó a rendszeren található összes hálózati csatolót kezelheti. Nem korlátozhatja a felhatalmazott felhasználó jogait, hogy csak bizonyos csatolókat módosíthasson.

Az RBAC tartomány szolgáltatása segítségével korlátozható a hozzáférés a felhatalmazott felhasználókra. A felhasználók és a rendszer erőforrásai címkékkel, úgynevezett tartományokkal vannak jelölve, és a specifikus hozzáférési szabályok határozzák meg a felhasználók hozzáférését az erőforrásokhoz.

Meghatározások

A hozzáférési szabályokkal kapcsolatos meghatározások:

alany: Az alany olyan entitás, amely hozzáférést kér egy objektumhoz. Alany például egy folyamat.

objektum: Az objektum olyan entitás, amely értékes információkat tartalmaz. Objektumok például a fájlok, eszközök és hálózati portok.

tartomány: A tartomány olyan kategória, amelybe egy entitás tartozik. Amikor egy entitás egy tartományba tartozik, akkor az entitás hozzáférés-felügyeletét a hozzáférési szabályok határozzák meg az alábbiak szerint:

Hozzáférési szabályok

- Egy alany akkor férhet hozzá egy objektumhoz, ha az összes olyan tartománnyal rendelkezik, amelybe az objektum tartozik. Ez meghatározza, hogy az alany tartományainak listája bővebb, mint az objektum tartományainak készlete. Ez az alapértelmezett viselkedés.
- Egy alany akkor férhet hozzá egy objektumhoz, ha rendelkezik az objektumnak legalább egy tartományával. Vagyis az alanynak és az objektumnak van legalább egy közös tartománya. Ez a viselkedés az objektum biztonsági kapcsolóitól függ.
- Egy objektum megtagadhatja a hozzáférést bizonyos tartományokhoz. Ha egy objektum meghatároz egy tartománykészletet mint ütközési készletet, és az alany tartományainak egyike az ütközési készlet része, akkor az objektum megtagadhatja a hozzáférést az alany számára.

Tartomány-adatbázis

A rendszer által támogatott tartományokat egy konfigurációs fájlban kell tárolni az /etc/security/domains alatt. A fájl egy szakaszának formátuma az alábbi:

```
domain-name:
id = <szám>
dfltmsg = <Üzenet>
msgcat = <Üzenetkatalógus>
msgset = <Üzenetkészlet a katalógusban>
msgnum = <Üzenetazonosító a katalógusban>
```

Az adatbázis a **mkdom** és a **chdom** paranccsal kezelhető. Az **lsdom** paranccsal jeleníthető meg az adatbázis. A bejegyzések az **rmdom** paranccsal törölhetők.

Az adatbázis bejegyzései mindaddig nem lépnek érvénybe, amíg le nem tölti az adatbázist a kernelbe a **setkst** paranccsal.

Legfeljebb 1024 tartomány támogatott a rendszeren, és a tartományazonosító (ID attribútum) lehetséges legnagyobb értéke 1024.

Tartományhoz rendelt objektumok

Ahhoz, hogy tartományt rendeljen egy objektumhoz, határozza meg azt a tartományhoz rendelt objektumok adatbázisában. A rendszeren található összes entitás tartományait az /etc/security/domobjs alatti konfigurációs fájl tárolja. A fájl egy szakaszának formátuma az alábbi, amely példa egy tartomány hozzárendelésére egy objektumhoz:

```
/dev/hrvg:
domains=HR,IT
conflictsets=payroll
objtype=device
secflags=FSF_DOM_ANY
```

domains: Meghatározza mindazokat a tartományokat, amelyek számára engedélyezett az objektum elérése. Tartományok például: IT, HR és Payroll.

objtype: A tartományhoz hozzárendelt objektum típusát jelzi. Az objektumtípusok: device, file, netint és netport.

conflict sets: Azt jelzi, hogy amennyiben az alany az ebben az attribútumban felsorolt tartományok valamelyikébe tartozik, akkor nem megengedett számára az objektum elérése.

secflags: Ez a kapcsoló adja meg az objektum speciális tulajdonságait. A kapcsolók beállítása **FSF_DOM_ANY** vagy **FSF_DOM_ALL** lehet. Ha a kapcsoló **FSF_DOM_ANY**, akkor az alany abban az esetben érheti el az objektumot, ha tartalmazza a domains attribútum felsorolásában megadott tartományok valamelyikét. Ha a kapcsoló **FSF_DOM_ALL**, akkor a felsorolásban szereplő összes tartományt ki kell elégítenie az alanynak az objektum eléréséhez. Ha nincs megadva érték, akkor az alapértelmezett **FSF_DOM_ALL** érték kerül felhasználásra. A **secflag** csak az objektum domains attribútumának viselkedésére van hatással.

Lehetőség van tartományok hozzárendelésére a fájlrendszerek fájljaihoz. Alapértelmezésben az objektum összes tartományának a folyamat tartományainak részhalmazát kell képeznie ahhoz, hogy a folyamat hozzáférhessen az objektumhoz.

1. Eszközök: Minden eszköz (beleértve a fájlrendszereket is) hozzárendelhető egy tartományhoz. A tartományellenőrzésekre a kezelési tevékenységek, például az eszköz beállítása közben kerül sor.

   ```
   /dev/hrvg:
   domains=HR,IT
   conflictsets=payroll
   objtype=device
   secflags=FSF_DOM_ANY
   ```

2. Hálózati csatolók: Amikor hálózati csatolók (például: en0) vannak hozzárendelve a tartományhoz, akkor a kezelési tevékenységek, például a csatoló leállítása megköveteli, hogy a csatoló átmenjen a tartományellenőrzéseken.

   ```
   en0:
   domains=NETIF,ADMIN
   objtype=netint
   flags=FSF_DOM_ALL
   ```

3. Hálózati portok: A TCP és az UDP portok hozzárendelhetők a tartományhoz. Tartományellenőrzésekre kerül sor, amikor egy alkalmazás megpróbál csatlakozni egy porthoz.

   ```
   TCP_<port#>:
   domains=NETIF,ADMIN
   type=netport
   flags=FSF_DOM_ALL
   ```

4. Folyamatok: A folyamat örökli annak a felhasználónak a tartományait, akinek a nevében a folyamat fut. Amikor a felhasználó bejelentkezik, akkor a felhasználó héj folyamata a felhasználó tartományaival rendelkezik. A tartományok beállításakor a folyamat ezen tartományai egész élete során megmaradnak. A folyamat tartományait nem módosíthatja egy akármilyen felhasználói felület vagy rendszerhívás. Az egyetlen folyamat, amely beállíthatja a tartományokat, a bejelentkezési folyamat. A folyamatok nem rendelkeznek **conflict set** és **secflags** attribútummal.

Aktuális korlátozások

Az alábbi elemek az aktuális tartomány RBAC szolgáltatás korlátozásai:

- A tartománykonfigurációs fájlok jelenleg csak a helyi rendszeren támogatottak, az LDAP kiszolgálón nem.

- Az RBAC tartományok nem támogatottak AIX munkaterhelési partíciókban (WPAR).
- Nem alkalmazhat RBAC tartományokat átmeneti fájlokra.

Kiterjesztett RBAC követelmény

A tartomány RBAC a Kiterjesztett RBAC után jön létre, és működéséhez szükséges, hogy a Kiterjesztett RBAC engedélyezett legyen a rendszeren.

Kernel biztonsági táblázatok

A tartomány-adatbázisban és a Tartomány-objektum adatbázisban meghatározott tartományok és Tartományhoz rendelt objektumok az után lépnek érvénybe, hogy letölti őket a kernelbe a **setkst** paranccsal. A két tábla a Kernel tartomány tábla (KDOMT) és a Kernel tartomány objektum tábla (KDOT).

A kernel biztonsági táblákkal és a **setkst** paranccsal kapcsolatos további részleteket az AIX biztonsági kézikönyv szerepalapú hozzáférés-felügyelet (RBAC) témaköre tartalmaz.

Tartomány parancsok

Az alábbi táblázat felsorolja mindazokat a tartomány RBAC szolgáltatással kapcsolatos parancsokat, amelyek rendelkezésre állnak az AIX operációs rendszeren a tartomány RBAC keretrendszer kezeléséhez és használatához:

Parancs	Leírás
mkdom	Létrehoz egy új tartományt
lsdom	Megjeleníti a tartomány attribútumait
rmdom	Eltávolítja a tartományt
chdom	Módosítja a tartomány attribútumait
setsecattr	Beállítja a Tartomány-objektum adatbázis biztonsági attribútumait
lssecattr	Megjeleníti a Tartomány-objektum adatbázis biztonsági attribútumait
rmsecattr	Eltávolítja a Tartomány-objektum adatbázis meghatározását
setkst	Elküldi a tartomány RBAC felhasználói szintű adatbázisok bejegyzéseit a Kernel biztonsági táblázatokba

Tartomány RBAC kapcsolódó fájlok

Az alábbi táblázat felsorolja mindazokat az RBAC szolgáltatással kapcsolatos fájlokat, amelyek rendelkezésre állnak az AIX operációs rendszeren az adatbázis információk beállításhoz és tárolásához:

Fájl	Leírás
/etc/security/domains	Tartomány-adatbázis
/etc/security/domobjs	Tartomány-objektum adatbázis

Tartományok használata

Tartományok meghatározása: A tartományokat a **mkdom** paranccsal lehet meghatározni a Tartomány-adatbázisban.

```
mkdom id=24 HR
```

Tartományok hozzárendelése: A tartományok hozzárendelhetők entitásokhoz, például felhasználókhoz, fájlokhoz, eszközökhöz, hálózati portokhoz és csatolóhoz. A felhasználók kivételével az összes entitás támogatja az ütközési beállításokat és a biztonsági kapcsolókat (**secflags**).

Felhasználók: A felhasználókat a **chuser** és a **chsec** paranccsal lehet hozzárendelni a tartományokhoz.

Szintaxis:

```
chuser domains = <tartományok vesszővel elválasztott listája> felhasználónév
```

Példa:

```
chuser domains=INET john
```

A bejelentkezés során aktiválódnak a felhasználóhoz hozzárendelt tartományok. Ha a tartományok megváltoztak, miközben a munkamenet aktív volt, akkor újra be kell jelentkeznie, hogy érvénybe lépjenek az új tartományok.

Objektumok: Ahhoz, hogy korlátozza az objektumok elérését a tartományokon keresztül, meg kell határoznia az objektumot a Tartomány-objektum adatbázisban a **setsecattr** paranccsal.

Szintaxis:

```
setsecattr -o domains=<megengedett tartományok vesszővel elválasztott listája>
conflictsets=<tiltott tartományok vesszővel elválasztott listája>
secflags=<FSF_DOM_ALL vagy FSF_DOM_ANY>
objtype=<file vagy device vagy netint vagy netport>
object-path
```

Példa:

```
setsecattr -o domains=INET,WEB conflictsets=DB secflags=FSF_DOM_ANY objtype=netint en0
```

Hozzáférés felügyeleti listák

Az ACL általában hozzáférés felügyeleti bejegyzéseknek (ACE) nevezett bejegyzések sorozatából áll. Minden ACE egy felhasználónak az adott objektumhoz fűződő elérési jogait írja le.

Hozzáférési kísérletkor az operációs rendszer az objektumhoz rendelt ACL lista segítségével megállapítja, hogy a felhasználó jogosult-e erre. Ezek az ACL listák és az ehhez kapcsolódó hozzáférési ellenőrzések alkotják az AIX által támogatott Korlátlan hozzáférés felügyelet (DAC) mechanizmus magját.

Az operációs rendszer számos olyan rendszerobjektum típust támogat, amelyek lehetővé teszik a felhasználói folyamatok számára az információk tárolását és kommunikálását. A hozzáférés felügyelet hatáskörébe vont legfontosabb objektumtípusok a következők:

- Fájlok és könyvtárak
- megnevezett csővezetékek
- IPC objektumokok, például üzenetsorok, osztott memóriaszegmensek és szemaforok

Az objektumok hozzáférés jogosultsági ellenőrzése rendszerhívás szinten kerül végrehajtásra az objektum első hozzáférésekor. Mivel a System V folyamatközi kommunikációs (SVIPC) objektumok állapot nélkül kerülnek hozzáférésre, ezért a rendszer minden hozzáférést ellenőriz. A fájlrendszer nevekkel rendelkező objektumoknál fel kell oldani a neveket a tényleges objektumokra. A nevek feloldhatók relatív (a folyamat munkakönyvtárára) vagy abszolút (a folyamat gyökér könyvtárára) módon. Minden névfeloldás a két könyvtár valamelyikének keresésével kezdődik.

A megítélés szerinti hozzáférés felügyeleti mechanizmus lehetővé teszi az információs erőforrások hatékony hozzáférés felügyeletét, és külön biztosítja az információk bizalmasságának és integritásának védelmét. A tulajdonosok által kezelt hozzáférés felügyeleti mechanizmusok csak annyira hatékonyak, amennyire a felhasználók hatékonnyá teszik. Minden felhasználónak tisztában kell lennie azzal, hogy a hozzáférés jogosultságok hogyan kerülnek kiosztásra vagy tiltásra illetve beállításra.

Például egy fájlrendszer objektumhoz (fájlhoz vagy könyvtárhoz) rendelt ACL érvényre juttatja a különböző felhasználók hozzáféréshez fűződő jogosultságait. Egy ilyen ACL segítségével akár különböző szintű (olvasási és írási) hozzáférési jogok kezelése is megvalósítható különböző felhasználók számára.

általában minden objektum rendelkezik egy tulajdonossal, és bizonyos esetekben beletartozik egy csoportba. Az objektum tulajdonosa felügyeli az objektum egyéni hozzáférési attribútumait. A tulajdonos attribútumai a létrehozó folyamat tényleges felhasználói azonosítójára vannak beállítva.

Az alábbi lista a különböző típusú objektumok közvetlen hozzáférés felügyeleti attribútumait tartalmazza:

Tulajdonos
A System V folyamatközi kommunikációs (SVIPC) objektumok tulajdonosát a létrehozó és a tulajdonos is módosíthatja. Az SVIPC objektumot rendelkeznek létrehozóval, aki a tulajdonos összes jogával rendelkezik (beleértve a hozzáférés jogosultságot). A létrehozót nem lehet módosítani, még root jogosultsággal sem.

A SVIPC objektumot a rendszer a létrehozó folyamat tényleges csoport azonosítójára inicializálja. A fájlrendszer-objektumoknál a rendszer a hozzáférés felügyeleti attribútumokat a létrehozó folyamat tényleges csoport azonosítójára vagy a szülőkönyvtár csoport azonosítójára inicializálja (ez a szülőkönyvtár csoport öröklés jelzője alapján kerül meghatározásra).

Csoport
Az objektum tulajdonosa megváltoztathatja a csoportot. Az új csoport lehet a létrehozási folyamat tényleges csoportazonosítója, vagy pedig a szülőkönyvtár csoportazonosítója. (Ahogy fent láthattuk, az SVIPC objektumoknak van egy társított létrehozó csoportja, amelyet nem lehet módosítani, és amely megosztja az objektum csoport hozzáférési jogosultságait.)

Mód A **chmod** parancs (numerikus módban oktális jelölésekkel) állítja be a jogosultságokat és attribútumokat. A parancs által meghívott **chmod** szubrutin letiltja a kiterjesztett jogosultságokat. Ha a **chmod** parancsot numerikus módban használja egy ACL-llel rendelkező fájlon, akkor a kiterjesztett jogosultságok letiltásra kerülnek. A **chmod** parancs szimbolikus módja nem tiltja le a kiterjesztett jogosultságokat, de az NSF4 ACL típus kiterjesztett ACL-jeit igen. A numerikus és szimbolikus módról a **chmod** parancsnál talál további információkat.

Az operációs rendszer számos objektuma, például a socket-ek vagy a fájlrendszer objektumok esetében az alobjektumokhoz külön ACL listák tartoznak. Az ilyen objektumok ACL listáinak részletei típusonként eltérnek.

Az AIX a hagyományosan támogatott módbiteket használ a fájlrendszer-objektumok elérésének szabályozására. Ezenkívül a módbitekhez az ACL egy egyedi formáját támogatta. Ez az ACL tartalmazta az alapvető módbiteket, ezenkívül lehetővé tette további ACE bejegyzések létrehozását is, amelyek az egyes felhasználók és csoportok hozzáférési jogait határozták meg. Ez a klasszikus típusú ACL viselkedés továbbra is támogatott marad, AIXC ACL típus elnevezéssel.

Megjegyzendő, hogy a fájlrendszer objektumokon megvalósított ACL listák támogatása az alapul szolgáló fizikai fájlrendszertől (PFS) függ. A fizikai fájlrendszernek meg kell értenie az ACL adatokat, és képesnek kell lennie a különböző felhasználók jogosultságainak tárolására, visszakeresésére és betartatására. Egyes fájlrendszerek egyáltalán nem támogatják az ACL listákat (csak az alapvető módbiteket), míg más fájlrendszerek többféle ACL listát is támogathatnak. A fájlrendszerek némelyike AIX alatt kiterjesztésre került több ACL típus támogatásával. A JFS2 és a GPFS például az NFS 4 protokollra épülő ACL listákat is támogatja. Ennek az ACL-nek a neve AIX rendszeren NFS4 ACL típus. Ez az ACL típus az NFS 4 protokoll specifikációjában szereplő ACL meghatározás legnagyobb részét megvalósítja. Az AIXC ACL típussal összehasonlítva a hozzáférés felügyelet finomabb megvalósítását teszi lehetővé, és olyan képességekkel rendelkezik, mint például az öröklődés.

Több hozzáférés felügyeleti lista típus keretrendszer támogatása

Az 5.3.0 változattól kezdődően az AIX támogat egy olyan infrastruktúrát, amely az operációs rendszerben a különböző fájlrendszer-objektumokhoz különböző ACL típusok használatát teszi lehetővé.

Ez az infrastruktúra az adott objektumhoz tartozó ACL típusától független, egységes felületet biztosít az ACL-ek kezeléséhez. A keretrendszer az alábbi összetevőket tartalmazza:

ACL adminisztrációs parancsok
Ilyen parancsok például az **aclget**, az **aclput**, az **acledit**, az **aclconvert** és az **aclgetttypes**. Ezek a parancsok olyan könyvtár illesztőket hívnak meg, amelyek ACL típusonként változó modulokat hívnak meg.

ACL könyvtárillesztők
A hozzáférés felügyeleti lista könyvtár illesztők külső felületetet képeznek az olyan alkalmazások számára, amelyeknek el kell érniük a hozzáférés felügyeleti listákat.

ACL típustól függő dinamikusan betölthető ACL modulok
Az AIX operációs rendszer több ACL típus-specifikus modult biztosít AIX klasszikus ACL-ekhez (**AIXC**) és NFS4 ACL-ekhez (**nfs4**).

Bináris kompatibilitás:

Nincsenek kompatibilitási problémák azon alkalmazások esetében, amelyek a meglévő JFS2 fájlrendszereken futnak, a meglévő AIX ACL-ekkel vagy azok nélkül.

Azonban előfordulhat, hogy az alkalmazásoknak nem sikerül a sokkal szigorúbb (például NFS4) ACL listával rendelkező fájlrendszer objektumok elérése. A fájl meglétének ellenőrzése például olvasási szintű hozzáférést követel meg az NFS4 ACL szerint.

Az AIX operációs rendszeren támogatott hozzáférés-felügyeleti lista típusok
Az AIX operációs rendszer jelenleg az AIXC és NFS4 ACL típusokat támogatja.

Mint korábban már volt róla szó, az AIX egy olyan keretrendszert is tartalmaz, melynek segítségével ez kibővíthető az alapul szolgáló fájlrendszer által támogatott tetszőleges ACL típussal. Tartsa szem előtt, hogy a JFS2 fizikai fájlrendszer natívan támogatja az NFS4 ACL-t abban az esetben, ha a fájlrendszer példányt "Kiterjesztett attribútumok 2-es változat" képességgel hozták létre.

AIXC hozzáférés felügyeleti lista:

Az AIXC hozzáférés felügyeleti lista típus az AIX 5.3.0 előtti kiadásain támogatott ACL típus viselkedését hivatott képviselni. Az AIXC ACL-ek alapjogosultságokat és kiterjesztett jogosultságokat tartalmaznak.

Az AIXC hozzáférés felügyeleti lista (ACL) típus az AIX 5.3.0 előtti kiadásain támogatott ACL típus viselkedését hivatott képviselni. Az AIXC ACL-ek alapjogosultságokat és kiterjesztett jogosultságokat tartalmaznak. A JFS2 fájlrendszer maximálisan 4 Kb-ot engedélyez az AIXC hozzáférés felügyeleti listák számára.

AIXC ACL alapjogosultságainak beállítása

Az alapjogosultságok általában a fájltulajdonoshoz, fájlcsoporthoz és egyéb felhasználókhoz rendelt hagyományos fájlhozzáférési módok. A hozzáférési módoki: olvasás (r), írás (w) és végrehajtás/keresés (x).

A hozzáférés felügyeleti listákban az alapjogosultságok a következő formában vannak megadva, a *Mode* paramétert rwx jelöli (a meg nem határozott jogosultságokat kötőjel (-) jelzi):

```
base permissions:
   owner(name): Mode
   group(group): Mode
   others: Mode
```

AIXC ACL attribútumainak beállítása

Az AIXC hozzáférés felügyeleti listákhoz a következő attribútumokat lehet hozzáadni:

setuid (SUID)
Felhasználói azonosító beállítása mód bit. Ez az attribútum állítja be a fájl tulajdonos azonosítóját a folyamat tényleges és mentett felhasználói azonosítójára a végrehajtáskor.

setgid (SGID)
Csoport azonosító beállítása mód bit. Ez az attribútum állítja be a fájl csoport azonosítóját a folyamat tényleges és mentett csoport azonosítójára a végrehajtáskor.

savetext (SVTX)
A könyvtáraknál azt jelzi, hogy csak a fájl tulajdonosok hozhatnak létre illetve törölhetnek fájl hivatkozásokat az adott könyvtárban.

Az attribútumok az alábbi formában kerülnek hozzáadásra:

```
attribútumok: SUID, SGID, SVTX
```

AIXC Access ACL kiterjesztett jogosultságainak beállítása

A kiterjesztett jogosultságok lehetővé teszik a fájl tulajdonosa számára, hogy még pontosabban meghatározza a fájlhoz való hozzáférést. A kiterjesztett jogosultságok adott felhasználók, csoportok vagy felhasználó és csoport kombinációk engedélyezésével, letiltásával vagy meghatározott hozzáférési módokkal való ellátásával módosítják az alap fájljogosultságokat (tulajdonos, csoport, egyebek). A jogosultságokat kulcsszavakkal lehet megadni.

A **permit**, **deny** és **specify** kulcsszavak meghatározása:

permit Megadott hozzáférést ad a fájlhoz az adott felhasználó vagy csoport számára.

deny Korlátozza egy adott felhasználó vagy csoport megadott hozzáférését a fájlhoz.

specify Pontosan megadja egy adott felhasználó vagy csoport fájlhozzáférését.

Ha egy felhasználót a **deny** vagy **specify** kulcsszó egy adott hozzáféréstől eltilt, akkor semmilyen egyéb bejegyzés nem bírálhatja felül ezt a hozzáférés tiltást.

Az **enabled** kulcsszót meg kell adni az ACL-ben ahhoz, hogy a kiterjesztett jogosultságok életbe lépjenek. Az alapértelmezett érték a **disabled** kulcsszó.

A kiterjesztett jogosultságok formátuma a hozzáférés felügyeleti listákban:

```
extended permissions:
  enabled | disabled
    permit   Mode  UserInfo...
    deny     Mode  UserInfo...
    specify  Mode  UserInfo...
```

Írjon külön sorban minden egyes **permit**, **deny** és **specify** bejegyzést. A *Mode* paramétert **rwx** jelöli (a meg nem határozott jogosultságokat kötőjel (-) helyettesíti). A *UserInfo* paramétert u:Felhasználónév, g:Csoportnév, vagy a u:Felhasználónév és g:Csoportnév veszővel elválasztott kombinációja jelöli.

Megjegyzés: Egy folyamathoz csak egyetlen felhasználói azonosító tartozhat, ezért ha több felhasználónevet ad meg egy bejegyzésben, akkor az adott bejegyzés nem játszhat szerepet a hozzáférés felügyeleti döntés során.

AIXC ACL szöveges ábrázolása

A következő szakasz egy AIXC hozzáférés felügyeleti lista szöveges megjelenését mutatja:

```
Attributes:  { SUID | SGID | SVTX }
Base Permissions:
   owner(name): Mode
   group(group): Mode
   others: Mode
Extended Permissions:
   enabled | disabled
     permit   Mode  UserInfo...
     deny     Mode  UserInfo...
     specify  Mode  UserInfo...
```

AIXC ACL bináris formátuma

Az AIXC hozzáférés felügyeleti lista bináris formátuma az /usr/include/sys/acl.h fájlban van definiálva, és az aktuális AIX kiadásban megvalósításra került.

AIXC ACL példa

Példa egy AIXC hozzáférés felügyeleti listára:

```
attributes: SUID
base permissions:
      owner(frank):  rw-
      group(system): r-x
      others: ---
extended permissions:
      enabled
        permit  rw-  u:dhs
        deny    r--  u:chas, g:system
        specify r--  u:john, g:gateway, g:mail
        permit  rw-  g:account, g:finance
```

Az ACL bejegyzések leírása a következő:

- Az első sor azt jelzi, hogy a **setuid** bit be van kapcsolva.
- Az alapjogosultságokat bevezető következő sor megadása nem kötelező.
- A következő három sor az alapjogosultságokat adja meg. A zárójelekben található tulajdonos- és csoportnevek csak információs célokat szolgálnak. Ezeknek a neveknek a módosítása nem módosítja a fájl tulajdonost vagy fájlcsoportot. Ezeket a fájlattribútumokat csak a **chown** és a **chgrp** paranccsal lehet módosítani.
- A kiterjesztett jogosultságokat bevezető következő sor megadása nem kötelező.
- A következő sor azt jelzi, hogy a sor alatt következő jogosultságok engedélyezve vannak.
- Az utolsó négy sor a kiterjesztett bejegyzéseket tartalmazza. Az első kiterjesztett bejegyzés *dhs* olvasási (r) és írási (w) jogosultságot ad a fájlra.
- A második kiterjesztett bejegyzés megtiltja az olvasási (r) hozzáférést a *chas* felhasználónak de csak abban az esetben, ha tagja a *system* csoportnak.
- A harmadik kiterjesztett bejegyzés megadja, hogy a *john* felhasználó mindaddig olvasási (r) hozzáféréssel rendelkezik, amíg tagja a *gateway* és a *mail* csoportnak. Ha a *john* felhasználó nem tagja mindkét csoportnak, akkor ez a kiterjesztett jogosultság nem vonatkozik rá.
- Az utolsó kiterjesztett bejegyzés olvasás (r) és írás (w) jogosultságot ad azoknak a felhasználóknak, akik az *account* és *finance* csoportnak *is* tagjai.

  > **Megjegyzés:** A felügyelt objektumhoz hozzáférést kérő folyamatokra több kiterjesztett bejegyzés is vonatkozhat. A bejegyzések közül a korlátozó bejegyzések elsőbbséget élveznek az engedélyező módokkal szemben.

  A teljes szintaxis a *Commands Reference* kiadvány **acledit** paranccsal foglalkozó részében található.

NFS4 hozzáférés felügyeleti lista:

Az AIX az NFS4 hozzáférés felügyeleti lista (ACL) típust is támogatja.

Az NFS4 hozzáférés felügyeleti lista a hozzáférésfelügyeletet a *Hálózati fájlrendszer (NFS) 4. változat protokoll RFC 3530* dokumentumban meghatározott módon valósítja meg. A JFS2 fájlrendszer maximálisan 64 Kb-ot engedélyez az NFS4 hozzáférés felügyeleti listák számára.

Csak az NFS V4 kliens támogatja az NFS V4 ACL listákat. A Cachefs és a Proxy nem támogatja az NFS V4 ACL listákat.

NFS4 ACL szöveges ábrázolása

Az NFS V4 hozzáférés felügyeleti lista hozzáférés felügyeleti bejegyzések (ACE) olyan listája, ahol minden sorban egy hozzáférés felügyeleti bejegyzés szerepel. A hozzáférés felügyeleti bejegyzéseknek négy eleme van a következő formátumban.

```
IDENTITY    ACE_TYPE      ACE_MASK     ACE_FLAGS

ahol:
IDENTITY => Formátuma 'IDENTITY_type:(IDENTITY_name vagy IDENTITY_ID vagy IDENTITY_who):'
         ahol:
         IDENTITY_type => Az alábbi azonosság típusok valamelyike:
                  u : felhasználó
                  g : csoport
                  s : speciális ki karaktersorozat (az IDENTITY_who-nak egy speciális ki-nek kell lennie)
                          IDENTITY_name => felhasználó/csoportnév
                          IDENTITY_ID   => felhasználó/csoportazonosító
                          IDENTITY_who  =>  speciális ki karaktersorozat (például OWNER@, GROUP@, EVERYONE@)
ACE_TYPE => Az alábbi hozzáférés felügyeleti bejegyzés típusok valamelyike:
                a : engedélyezés
                d : tiltás
                l : figyelmeztetés
                u : megfigyelés
ACE MASK =>  Az alábbi maszk érték kulcsok közül néhány elválasztó nélkül:
                r : READ_DATA   vagy LIST_DIRECTORY
                w : WRITE_DATA      vagy ADD_FILE
                p : APPEND_DATA     vagy ADD_SUBDIRECTORY
                R : READ_NAMED_ATTRS
                W : WRITE_NAMED_ATTRS
                x : EXECUTE         vagy SEARCH_DIRECTORY
                D : DELETE_CHILD
                a : READ_ATTRIBUTES
                A : WRITE_ATTRIBUTES
                d : DELETE
                c : READ_ACL
                C : WRITE_ACL
                o : WRITE_OWNER
                s : SYNCHRONIZE
ACE_FLAGS (Nem kötelező) =>  Az alábbi attribútumkulcsok közül néhány elválasztó nélkül:
                fi : FILE_INHERIT
                di : DIRECTORY_INHERIT
                oi : INHERIT_ONLY
                ni : NO_PROPAGATE_INHERIT
                sf : SUCCESSFUL_ACCESS_ACE_FLAG
                ff : FAILED_ACCESS_ACE_FLAG
```

Megjegyzés: A SYNCHRONIZE Ace_Mask érték kulcsnál s, az AIX nem végez semmilyen műveletet ezzel a értékkulccsal. Az AIX operációs rendszer tárolja és megőrzi az s értékkulcsot, de ennek az értékkulcsnak nincsen semmilyen jelentése az AIX rendszer számára.

Ha a WRITE_OWNER Ace_Mask értéke Ace_Type allow, akkor a felhasználók a fájl tulajdonjogát csak magukra módosíthatják.

A fájl törlése két ACE értéktől függ, a törölni kívánt objektum DELETE bejegyzésétől és a szülőkönyvtár DELETE_CHILD bejegyzésétől. Az AIX operációs rendszer a felhasználó számára kétféle viselkedésmódot biztosít. *Biztonságos* módban a DELETE az AIXC ACL-ekhez hasonlóan működik. *Kompatibilitás* módban a DELETE az NFS4 ACL-ek más fő megvalósításához hasonlóan működik. A kompatibilitási mód bekapcsolásához használja a **chdev** parancsot a következőképp:

```
chdev -l sys0 -a nfs4_acl_compat=compatible
```

A **chdev** parancs futtatása után a konfigurációmódosítás életbe lépéséhez a rendszert újra kell indítani.

Ha a rendszet a két mód között váltogatja oda és vissza, akkor figyelnie kell arra, hogy az AIX operációs rendszer által biztonságos módban előállított NFS4 ACL-eket lehet, hogy nem fogadják el más platformok még akkor sem, ha a rendszer vissza lett váltva kompatibilitási módba.

```
Példa:
        u:user1(aa@ibm.com):     a    rwp    fidi
        *s:(OWNER@):             d    x      dini          * Ez a sor egy megjegyzés
        g:staff(jj@jj.com):      a    rx
        s:(GROUP@):              a    rwpx   fioi
        u:2:                     d    r     di             * Felhasználó tároló (uid=2)
        g:7:                     a    ac    fi             * Csoport biztonság (gid=7)
        s:(EVERYONE@):           a    rca   ni
```

NFS4 ACL bináris formátuma

Az NFS4 ACL bináris formátum a /usr/include/sys/acl.h fájlban van meghatározva, és az aktuális AIX kiadásbanvan megvalósítva.

NFS4 ACL példa

Az alábbi példa egy könyvtárra alkalmazott NFS4 hozzáférés felügyeleti listát mutat be (például j2eav2/d0 könyvtárra):

```
s:(OWNER@):         a         rwpRWxDdo         difi         * 1st   ACE
s:(OWNER@):         d         D                 difi         * 2nd  ACE
s:(GROUP@):         d         x                 ni           * 3rd   ACE
s:(GROUP@):         a         rx                difi         * 4th   ACE
s:(EVERYONE@):      a         c                 difi         * 5th   ACE
s:(EVERYONE@):      d         C                 difi         * 6th   ACE
u:user1:            a         wp                oi           * 7th   ACE
g:grp1:             d         wp                             *  8th  ACE
u:101:              a         C                              * 9th   ACE
g:100:              d         c                              * 10th  ACE
```

Az ACL bejegyzések az alábbiak szerint vannak leírva:

- Az első hozzáférés felügyeleti bejegyzés azt jelzi, hogy a tulajdonosnak a következő jogosultságai vannak a /j2eav2/d0 könyvtárra és az összes olyan alkönyvtárra, amely a hozzáférés felügyeleti lista alkalmazása után került létrehozásra:
  - READ_DATA (= LIST_DIRECTORY)
  - WRITE_DATA (=ADD_FILE)
  - APPEND_DATA (= ADD_SUBDIRECTORY)
  - READ_NAMED_ATTR
  - WRITE_NAMED_ATTR
  - EXECUTE (=SEARCH_DIRECTORY)
  - DELETE_CHILD
  - DELETE
  - WRITE_OWNER
- A második hozzáférés felügyeleti bejegyzés azt jelzi, hogy a tulajdonosnak nincs DELETE_CHILD jogosultsága (a /j2eav2 könyvtárban létrehozott fájlok vagy alkönyvtárak törlése), de az első hozzáférés felügyeleti bejegyzés engedélyezi a DELETE_CHILD jogosultságot, így a tulajdonos törölheti a fájlokat és alkönyvtárakat.
- A harmadik hozzáférés felügyeleti bejegyzés azt jelzi, hogy az objektum (/j2eav2/d0) csoportjának tagjai nem rendelkeznek EXECUTE (=SEARCH_DIRECTORY) jogosultsággal, de a tulajdonos az első hozzáférés felügyeleti bejegyzés miatt rendelkezik ezzel a jogosultsággal. Ezt a hozzáférés felügyeleti bejegyzést nem lehet örökíteni a leszármazottakra, mert a NO_PROPAGATE_INHERIT kapcsoló meg van adva. A hozzáférés felügyeleti bejegyzés csak a /j2eav2/d0 könyvtárra és a közvetlenül alatta található leszármazott fájlokra és alkönyvtárakra vonatkozik.
- A negyedik hozzáférés felügyeleti bejegyzés azt jelzi, hogy az objektum (/j2eav2/d0) csoportjának minden tagja rendelkezik READ_DATA (= LIST_DIRECTORY) és EXECUTE (=SEARCH_DIRECTORY) jogosultsággal a /j2eav2/d0 könyvtárhoz és minden leszármazottjához. Ugyanakkor a harmadik hozzáférés felügyeleti bejegyzés miatt a csoport tagjai (kivéve a tulajdonost) nem rendelkeznek EXECUTE (=SEARCH_DIRECTORY) jogosultsággal a /j2eav2/d0 könyvtárhoz és annak közvetlen leszármazott fájljaihoz és alkönyvtáraihoz.

- Az első hozzáférés felügyeleti bejegyzés azt jelzi, hogy mindenki rendelkezik READ_ACL jogosultsággal a /j2eav2/d0 könyvtárhoz és az összes olyan leszármazott könyvtárhoz és fájlhoz, amely a hozzáférés felügyeleti lista alkalmazása után került létrehozásra.
- A hatodik hozzáférés felügyeleti bejegyzés azt jelzi, hogy senki nem rendelkezik WRITE_ACL jogosultsággal a /j2eav2/d0 könyvtárhoz és annak leszármazottaihoz. Az NFS4 hozzáférés felügyeleti listáknál a tulajdonos mindig rendelkezik WRITE_ACL jogosultsággal a fájlokhoz és könyvtárakhoz.
- A hetedik hozzáférés felügyeleti bejegyzés azt jelzi, hogy a user1 felhasználó rendelkezik WRITE_DATA (=ADD_FILE) és APPEND_DATA (= ADD_SUBDIRECTORY) jogosultsággal a /j2eav2/d0 könyvtár összes leszármazottjához, de magához a /j2eav2/d0 könyvtárhoz nem.
- A nyolcadik hozzáférés felügyeleti bejegyzés azt jelzi, hogy a grp1 csoport tagjai nem rendelkeznek a WRITE_DATA (=ADD_FILE) és az APPEND_DATA (= ADD_SUBDIRECTORY) jogosultságokkal. Ez a hozzáférés felügyeleti bejegyzés az első hozzáférés felügyeleti bejegyzés miatt annak ellenére nem vonatkozik a tulajdonosra, hogy a tulajdonos tagja a grp1 csoportnak.
- A kilencedik hozzáférés felügyeleti bejegyzés az jelzi, hogy az **UID 101** felhasználó rendelkezik WRITE_ACL jogosultsággal, de a hatodik hozzáférés felügyeleti bejegyzés miatt a tulajdonoson kívül senki nem rendelkezik WRITE_ACL jogosultsággal.
- A tizedik hozzáférés felügyeleti bejegyzés azt jelzi, hogy a **GID 100** azonosítóval rendelkező csoport tagjai nem rendelkeznek READ_ACL jogosultsággal, de az ötödik hozzáférés felügyeleti bejegyzés miatt a csoport tagjai rendelkezni fognak ezzel a jogosultsággal.

Hozzáférés felügyeleti lista kezelése

Hozzáférés felügyeleti listák (ACL) megjelenítéséhez és beállításához parancsokat használhat.

Az alkalmazásfejlesztők és az egyéb alrendszerek fejlesztői az ebben a szakaszban leírt hozzáférés felügyeleti lista könyvtár illesztőket és hozzáférés felügyeleti lista átalakító rutinokat használhatják.

ACL adminisztrációs parancsok

Az alábbi parancsokkal kezelheti a fájlrendszer objektumok hozzáférés felügyeleti listáit:

aclget A *FájlObjektum* hozzáférés felügyeleti listáját szabványos kimenetre írja, olvasható formában megjeleníti, vagy az *outAclFile* kimeneti fájlba írja.

aclput A szabványos bemenet vagy az *inAclFile* információi alapján beállítja a *FájlObjektum* hozzáférés felügyeleti listáját a fájlrendszeren.

acledit Megnyit egy szerkesztőt a megadott *FájlObjektum* hozzáférés felügyeleti listájának szerkesztéséhez.

aclconvert
A hozzáférés felügyeleti listát egyik típusról egy másik típusra alakítja. A parancs sikertelen lesz, ha az átalakítás nem támogatott.

aclgettypes
A fájlrendszer elérési út által támogatott hozzáférés felügyeleti lista típusokat keresi vissza.

ACL könyvtárillesztők

A hozzáférés felügyeleti lista könyvtár illesztők külső felületetet képeznek az olyan alkalmazások számára, amelyeknek el kell érniük a hozzáférés felügyeleti listákat. Az alkalmazások (beleértve a fenti általános ACL adminisztrációs parancsokat) nem hívják meg közvetlenül a nem dokumentált ACL rendszerhívásokat, hanem a könyvtár felületeken keresztül érik el az általános rendszerhívásokat és a típustól függő betölthető modulokat. Így az alkalmazásfejlesztőknek nem kell betölthető modulokat használniuk, és kevesebb visszamenőleges bináris kompatibilitási probléma merül fel a jövőbeni AIX kiadásoknál.

Az alábbi könyvtár illesztők rendszerhívásokat hívnak.

aclx_fget és aclx_get
Az **aclx_get** és **aclx_fget** függvények visszakeresik egy fájlrendszer objektum hozzáférés felügyeleti információit, és kiírják az **acl** által megadott memóriaterületre. Az **acl** méret és típus információi az ***acl_sz** és ***acl_type** értékekben kerülnek eltárolásra.

aclx_fput és aclx_put
Az **aclx_put** és **aclx_fput** függvények eltárolják az **acl** értékben megadott bemeneti fájl objektum hozzáférés felügyeleti információit. Ezek a függvények nem végeznek ACL típus átalakításokat. Az ACL típus átalakításához a hívónak kifejezetten meg kell hívnia az **aclx_convert** függvényt.

aclx_gettypes
Az **aclx_gettypes** függvény az adott fájlrendszeren támogatott ACL típusok listáját keresi vissza. Egy fájlrendszer egyszerre több ACL típust is támogathat. Minden fájlrendszer objektumhoz egyedi ACL típus van társítva, amely szerepel a fájlrendszer által támogatott ACL típusok listájában.

aclx_gettypeinfo
Az **aclx_gettypeinfo** függvény visszakeresi az elérési út által megadott fájlrendszer egyik ACL típusának jellemzőit és képességeit. Vegye figyelembe, hogy az ACL jellemzői általában adatszerkezet típusúak, amely minden egyes ACL típusnál más. Az AIXC és NFS4 hozzáférés felügyeleti listákhoz használt adatszerkezeteket egy külön dokumentum írja le.

aclx_print és aclx_printStr
Ez a két függvény a bináris formátumban megadott hozzáférés felügyeleti listát szöveges megjelenésre alakítja. Ezeket a függvényeket az **aclget** és az **acledit** parancsok hívják meg.

aclx_scan és aclx_scanStr
Ez a két funkció a szöveges megjelenésű hozzáférés felügyeleti listát bináris formátumba alakítja.

aclx_convert
A hozzáférés felügyeleti listát egyik típusról egy másik típusra alakítja. A függvényt a parancsokkal (**cp**, **mv** vagy **tar**) végzett implicit átalakításához használja a rendszer.

Hozzáférés felügyeleti lista átalakítása

Az ACL átalakítás lehetővé teszi az egy ACL típus átalakítását egy másik típusra. A különböző ACL típusok támogatása attól függ, hogy az adott fizikai fájlrendszer milyen ACL típusokat támogat. Nem minden fájlrendszer támogat minden ACL típust. Az egyik fájlrendszer például lehet hogy csak az AIXC ACL típusokat támogatja, míg egy másik az AIXC és NFS4 ACL típusokat. Az AIXC ACL-eket átmásolhatja a két fájlrendszer között, de az NFS ACL-ek második rendszerről első rendszerre másolásakor ACL átalakítást kell használnia. Az ACL átalakítás amennyire csak lehet megőrzi a hozzáférés felügyeleti információkat.

Megjegyzés: Az átalakítási folyamat csak közelítő, így hozzáférés felügyeleti információk vesztéséhez vezethet. Ezt figyelembe kell vennie az ACL átalakítások tervezésekor.

Az ACL átalakítás az AIX operációs rendszerben az alábbi infrastruktúrákkal támogatott:

Könyvtárrutinok
Ezek a rutinok és a felhasználói szintű ACL keretrendszer teszi lehetővé az ACL átalakításokat az egyik ACL típusról egy másik típusra.

aclconvert parancs
Ez a parancs ACL-eket alakít át.

aclput és acledit parancs
Ezek a parancsok ACL típusokat módosítanak.

cp és mv parancs
Ezek a parancsok több ACL típust kezelnek és végrehajtják a belső ACL átalakításokat is, ha erre szükség van.

backup parancs
Ez a parancs az ACL információkat ismert típusba és formátumra (AIXC ACL típus) alakítja, ha a mentést

örökölt formátumban kéri. Ha az ACL-t az ACL eredeti formátumában szeretné lekérni, akkor használja az **-U** kapcsolót. További információkat a mentés részben talál.

Minden ACL típus egyedi, és a hozzáférés felügyeleti maszkok finomítása igen különbözik az egyes ACL típusoknál. Az átalakító algoritmusok csak közelítőek, és nem egyenértékűek az ACL manuális átalakításával. Bizonyos esetekben az átalakítás nem pontos. Az NFS4 ACL-eket például nem lehet teljesen AIXC ACL-ekké alakítani, mert az ACL-ek 16 hozzáférési maszkot és öröklés szolgáltatást tartalmaznak, amelyeket az AIXC ACL típus nem támogat. Ha a hozzáférés felügyeleti információk elvesztése komoly problémákat okozhat, akkor ne használja az ACL átalakító szolgáltatásokat és illesztőket.

Megjegyzés: Az ACL átalakító algoritmusok nem nyilvánosak és bármikor módosításra kerülhetnek.

S bitek és hozzáférés felügyeleti listák

A **setuid** és **setgid** programokat az S bitek ACL-ekre alkalmazásával használhatja.

Setuid és setgid programok használata

A jogosultság bitek a legtöbb helyzetben hatékony hozzáférés felügyeletet biztosítanak az eszközökhöz. A még pontosabb hozzáférés felügyelet érdekében az operációs rendszer tartalmazza a **setuid** és **setgid** programokat.

Az AIX operációs rendszer az azonosságokat csak uid és gid kifejezésekkel határoz meg. Az azonosságot nem uid és gid terminológiával meghatározó ACL típusok az AIX azonosságmodellre vannak leképezve. Az NFS4 ACL típus például a felhasználói azonosságot a felhasználó@tartomány karaktersorozattal definiálja, a rendszer pedig ezt karaktersorozatot képezi le numerikus UID-kre és GID-kre.

A legtöbb programot a rendszer a programot meghívó felhasználó felhasználói vagy csoport jogosultságaival hajtja végre. A program tulajdonosa hozzáférési jogokat társíthat a programot meghívó felhasználóhoz, ha a programot **setuid** vagy **setgid** programnak állítja be. Ez azt jelenti, hogy a programnak be van állítva a setuid vagy setgid bitje a jogosultság mezőben. Amikor a programot egy folyamat hajtja végre, akkor a folyamat megkapja a program tulajdonosának hozzáférési jogait. A **setuid** programot a rendszer a tulajdonosának hozzáférési jogaival hajtja végre, míg a **setgid** program a saját csoportjának hozzáférési jogait kapja. Mindkét bit beállítható a jogosultsági mechanizmusnak megfelelően.

Bár a folyamat további hozzáférési jogokat kap, ezeket a jogokat a jogokkal rendelkező program felügyeli. Így a **setuid** és **setgid** programok lehetővé teszik az olyan felhasználók által programozott hozzáférés felügyeletek használatát, amelyekben a hozzáférési jogok indirekt módon kerülnek megadásra. A program megbízható alrendszerként működik, és vigyázza a felhasználó hozzáférési jogait.

Bár ezek a programok nagyon hatékonyan használhatók, biztonsági kockázatot is jelenthetnek, ha nincsenek megfelelően megtervezve. A program soha nem adhatja vissza a vezérlést a felhasználónak, amíg a tulajdonos hozzáférési jogaival rendelkezik, mivel ez a tulajdonos jogainak korlátlan használatát tenné lehetővé a felhasználó számára.

Megjegyzés: Az operációs rendszer biztonsági okokból nem támogatja a **setuid** és **setgid** program hívását parancsértelmező-fájlból.

S bitek alkalmazása a hozzáférés felügyeleti listákra

Az NFS4-hez hasonló ACL listák nem támogatják közvetlenül az S bitek használatát. Az NFS4 szabvány nem határozza meg, hogy ezeket a biteket milyen módon kell tárolni az ACL részeként. Az AIX operációs rendszer úgy közelítette meg a problémát, hogy S bitek kerülnek felhasználásra a hozzáférés ellenőrzés során, és értékeli az NFS4 ACL-hez kapcsolódó hozzáférési ellenőrzéseket. Az AIX operációs rendszerrel elérhető **chmod** parancs használható S bitek beállítására vagy visszaállítására fájlrendszer objektumokon hozzáférés-felügyeleti listával (ACL), például NFS4 esetén.

Adminisztrátori hozzáférési jogok

Az operációs rendszer kiváltságos hozzáférési jogokat biztosít a rendszeradminisztrációhoz.

A rendszerjogosultság a felhasználói- és csoport azonosítókon alapul. A 0 tényleges felhasználói- vagy csoport azonosítóval rendelkező felhasználókat a rendszer kiváltságos felhasználókként ismeri fel.

A 0 tényleges felhasználói azonosítóval rendelkező folyamatok a root felhasználói folyamatok, amelyek az alábbiakra képesek:

- Bármely objektum írása vagy olvasása
- Bármely rendszerfüggvény meghívása
- Bizonyos alrendszer vezérlő műveletek végrehajtása a **setuid-root** programokkal

A rendszert kétféle típusú jogosultsággal kezelheti: a **su** parancs jogosultsággal és a **setuid-root** program jogosultsággal. Az **su** parancs lehetővé teszi a programok meghívását root felhasználói folyamatként. Az **su** a rendszer kezelésének egy hatékony, ám nem túl biztonságos módja.

Egy program **setuid-root** programmá alakítása azt jelenti, hogy a program tulajdonosa a root felhasználó, és hogy a program setuid bitje be van állítva. A **setuid-root** program olyan adminisztrációs funkciókat biztosít, amelyeket a sima felhasználók a biztonság veszélyeztetése nélkül hajthatnak végre. A jogosultság a programba van beágyazva, és nem közvetlenül a felhasználóra vonatkozik. Az összes adminisztrációs funkció **setuid-root** programokba ágyazása bonyolult feladat lehet, de nagyobb biztonságot nyújt a rendszerkezelők számára.

Hozzáférés engedélyezése

Ha a felhasználó bejelentkezik egy fiókba (a **login** vagy **su** paranccsal), akkor a rendszer a felhasználói azonosítót illetve a csoport azonosítót hozzárendeli a felhasználó folyamataihoz. Ezek az azonosítók meghatározzák a folyamat hozzáférési jogait.

A 0 felhasználói azonosítóval rendelkező folyamatok a *root felhasználói folyamatok*. Ezek a folyamatok általában minden hozzáférési jogosultságot megkapnak. De ha a root felhasználói folyamat végrehajtási jogosultságot kér egy programra, akkor csak abban az esetben kapja meg a jogosultságot, ha a végrehajtás jogosultság legalább egy felhasználónak engedélyezve van.

AIXC hozzáférés felügyeleti listák hozzáférési jogosultságai

Az információs erőforrás tulajdonosa felelős azért, hogy a hozzáférési jogokat kezelje. Az erőforrásokat *jogosultság bitek* védik, amelyeket az objektum módja tartalmaz. A jogosultság bitek az objektum tulajdonosának, az objektum csoportjának illetve az egyéb alapértelmezett osztály hozzáférési jogait határozzák meg. Az operációs rendszer háromféle hozzáférési módot (írás, olvasás, végrehajtás) biztosít, amelyeket külön-külön lehet megadni.

A fájlok, könyvtárak, megnevezett csővezetékek és eszközök (különleges fájlok) számára a hozzáférést a rendszer a következőképpen adja meg:

- A rendszer a hozzáférés felügyeleti lista (ACL) minden egyes hozzáférés felügyeleti bejegyzésénél (ACE) összehasonlítja az azonosító listát a folyamat azonosítóival. Ha van egyezés, akkor a folyamat megkapja az adott bejegyzés jogosultságait és korlátozásait. A rendszer az ACL minden jogosultságának és megszorításának kiszámítja a logikai unióját. Ha a kérő folyamat nem felel meg az ACL egyik bejegyzésének sem, akkor az alapértelmezett bejegyzés jogosultságait és megszorításait kapja.
- Ha a kért hozzáférési mód engedélyezett (benne van a jogosultságok uniójában) és nincs korlátozva (nincs benne a korlátozások uniójában), akkor a rendszer megadja a hozzáférést. Ellenkező esetben a rendszer megtagadja a hozzáférést.

Az ACL azonosító listája csak akkor felel meg egy folyamatnak, ha a lista összes azonosítója megfelel kérő folyamat megfelelő tényleges azonosító típusainak. A USER típusú azonosító akkor megfelelő, ha egyezik a folyamat azonos, tényleges felhasználói azonosítójával, a GROUP típusú pedig ha egyezik a folyamat tényleges csoport azonosítójával vagy valamelyik kiegészítő csoport azonosítójával. Az alábbi azonosító listát tartalmazó ACE esetében:

```
USER:fred, GROUP:philosophers, GROUP:software_programmer
```

A folyamat akkor egyezik, ha a tényleges felhasználói azonosítója *fred*, és a következő csoportok tagja:
```
philosophers, philanthropists, software_programmer, doc_design
```

De nem egyezik a folyamat, ha a tényleges felhasználói azonosítója *fred*, és a következő csoportok tagja:
```
philosophers, iconoclasts, hardware_developer, graphic_design
```

Ne feledje, hogy az alábbi azonosító listát tartalmaz ACE mindkét folyamatnak megfelel:
```
USER:fred, GROUP:philosophers
```

Más szavakkal az ACE azonosító listájának feltételeivel mind rendelkeznie kell az adott folyamatnak a megfelelő hozzáférés megadásához.

Az objektumok hozzáférés jogosultsági ellenőrzése rendszerhívás szinten kerül végrehajtásra az objektum első hozzáférésekor. Mivel a System V folyamatközi kommunikációs (SVIPC) objektumok állapot nélkül kerülnek hozzáférésre, ezért a rendszer minden hozzáférést ellenőriz. A fájlrendszer nevekkel rendelkező objektumoknál fel kell oldani a neveket a tényleges objektumokra. A nevek feloldhatók relatív (a folyamat munkakönyvtárára) vagy abszolút (a folyamat gyökér könyvtárára) módon. Minden névfeloldás a két könyvtár valamelyikének keresésével kezdődik.

A megítélés szerinti hozzáférés felügyeleti mechanizmus lehetővé teszi az információs erőforrások hatékony hozzáférés felügyeletét, és külön biztosítja az információk bizalmasságának és integritásának védelmét. A tulajdonosok által kezelt hozzáférés felügyeleti mechanizmusok csak annyira hatékonyak, amennyire a felhasználók hatékonnyá teszik. Minden felhasználónak tisztában kell lennie azzal, hogy a hozzáférés jogosultságok hogyan kerülnek kiosztásra vagy tiltásra illetve beállításra.

NFS4 hozzáférés felügyeleti listák hozzáférési jogosultságai

A hozzáférési jogokat a WRITE_ACL jogosultsággal rendelkező felhasználók felügyelhetik. Az információforrás tulajdonosa mindig rendelkezik WRITE_ACL jogosultsággal. Az NFS4 hozzáférés felügyeleti listákkal rendelkező fájloknál és könyvtáraknál a hozzáférés engedélyezése a következő módon történik:

- A rendszer a hozzáférés felügyeleti bejegyzéseket sorrendben nézi végig, és csak azokat a hozzáférés felügyeleti bejegyzéseket dolgozza fel, amelyek rendelkeznek a kérővel azonos "ki" (azaz azonosság) jellemzővel. A rendszer nem ellenőrzi a kérő hitelesítési adatait, ha a hozzáférés felügyeleti bejegyzést a speciális EVERYONE@ azonossággal dolgozza fel.
- A hozzáférés felügyeleti bejegyzések addig kerülnek feldolgozásra, amíg a kérő hozzáférésének összes bitje engedélyezésre nem kerül. Ha egy bit már engedélyezve van, akkor a rendszer a későbbi hozzáférés felügyeleti bejegyzések feldolgozásakor már nem veszi figyelembe.
- Ha a kérő hozzáférésére vonatkozó bármely bit le van tiltva, akkor a hozzáférést a rendszer megtagadja, és a fennmaradó hozzáférés felügyeleti bejegyzéseket nem dolgozza fel.
- Ha a kérő hozzáférésének egyik bitje sincs engedélyezve és nincs több feldolgozásra váró hozzáférés felügyeleti bejegyzés, akkor a rendszer megtagadja a hozzáférést.

Ha a kért hozzáférést a hozzáférés felügyeleti bejegyzések megtagadják és a kérést kiadó felhasználó egy felettes vagy root felhasználó, akkor a rendszer általában megadja a hozzáférést. Ne feledje, hogy az objektum tulajdonosa mindig rendelkezik READ_ACL, WRITE_ACL, READ_ATTRIBUTES és WRITE_ATTRIBUTES jogosultsággal. További információk a hozzáférési jogosultságok algoritmusairól: “NFS4 hozzáférés felügyeleti lista” oldalszám: 122.

Hozzáférés felügyeleti lista hibáinak elhárítása

Az alábbi információk a hozzáférés felügyeleti lista (ACL) hibáinak elhárítását mutatják be.

NFS4 hozzáférés felügyeleti lista egy objektumhibás alkalmazásnál

Az NFS4 ACL-ek objektumon - például fájlon vagy könyvtáron - való beállításakor a hibaelhárításhoz használhatja a visszatérési kódot vagy a nyomkövetési szolgáltatást. Mindkét módszer az **aclput** és az **acledit** paranccsal keresi meg a problémát.

Visszatérési kód használata hibaelhárításhoz

A visszatérési kód megjelenítéséhez használja az echo $? parancsot az **aclput** parancs futtatása után. Az alábbi lista a visszatérési kódokat és azok magyarázatát mutatja be:

22 (EINVAL, a /usr/include/sys/errno.h fájlban van definiálva)
A hibakód lehetséges okai:

- Érvénytelen szöveges formátum a négy mező valamelyikében.
- Az bemeneti NFS4 ACL mérete nagyobb mint 64 Kb.
- Az ACL-t olyan fájlra alkalmazta, amely már rendelkezik legalább egy olyan hozzáférés felügyeleti bejegyzéssel, amelynek a maszkja w (WRITE_DATA) de nem p (APPEND_DATA) értékre, vagy p (APPEND_DATA) de nem w (WRITE_DATA) értékre van állítva.
- Az ACL-t olyan könyvtárra alkalmazta, amely már rendelkezik legalább egy olyan hozzáférés felügyeleti bejegyzéssel, amelynek a maszkja w (WRITE_DATA) de nem p (APPEND_DATA) értékre, vagy p (APPEND_DATA) de nem w (WRITE_DATA) értékre van állítva, illetve amelyben a fi kapcsoló (FILE_INHERIT) értékre van állítva.
- Van legalább egy olyan hozzáférés felügyeleti bejegyzés, amelyben az OWNER@ speciális **ki**-ként (**Azonosság**) van beállítva, és van egy vagy több hozzáférés felügyeleti bejegyzés maszk, amelyben a c (READ_ACL), C (WRITE_ACL), a (READ_ATTRIBUTE) és A (WRITE_ATTRIBUTE) jogosultságokat a d hozzáférés felügyeleti bejegyzés típus tiltja.

124 (ENOTSUP, a /usr/include/sys/errno.h fájlban van definiálva)
A hibakód lehetséges okai:

- Lehet hogy a speciális azonosság az egyik hozzáférés felügyeleti bejegyzésben nem a három lehetséges érték (OWNER@, GROUP@ vagy EVERYONE@) egyike.
- Van legalább egy olyan hozzáférés felügyeleti bejegyzés, amelynek típusa u (AUDIT) vagy l (ALARM).

13 (EACCES, a /usr/include/sys/errno.h fájlban van definiálva)
A hibakód lehetséges okai:

- Nincs jogosultsága az NFS4 ACE-t tartalmazó bemeneti fájl olvasásához.
- Nem kereshet a cél objektum szülő könyvtárában, mert a könyvtárhoz nem rendelkezik x (EXECUTE) jogosultsággal.
- Lehet hogy nincs jogosultsága az ACL írásához vagy módosításához. Ha az objektum már társítva van egy NFS4 ACL-hez, akkor győződjön meg róla, hogy rendelkezik jogosultsággal a C (WRITE_ACL) ACL maszkhoz.

Nyomkövetési szolgáltatás használata a hibaelhárításhoz

A probléma okának meghatározásához nyomkövetési jelentést is készíthet. Az alábbi példahelyzet bemutatja, hogyan lehet a nyomkövetéssel megkeresni az NFS4 ACL alkalmazásakor jelentkező hiba okát. A /j2v2/fájl1 fájl NFS4 ACL-je a következő:

```
s:(EVERYONE@):  a       acC
```

A bemeneti_acl_fájl bemeneti fájlban a következő ACL található:

```
s:(EVERYONE@):  a       rwxacC
```

Az alábbi lépések végrehajtásával a hibaelhárításhoz használhatja a nyomkövetési szolgáltatást:

1. Futtassa a trace, az **aclput** és a **trcrpt** parancsokat:

```
$ trace -j  478 -o trc.raw
$->!aclput -i input_acl_file -t NFS4 /j2v2/file1
$ ->quit
$ trcrpt trc.raw > trc.rpt
```

2. Elemezze a nyomkövetési jelentést. Ha az ACL egy fájlra vagy könyvtárra van alkalmazva, akkor a rendszer először ellenőrzi az ACL írásához vagy módosításához való hozzáférést, majd utána alkalmazza az ACL-t. A fájl az alábbiakhoz hasonló sorokat tartalmaz:

```
478 xxx  xxx  ACL ENGINE: chk_access entry: type=NFS4 obj_mode=33587200 size=68 ops=16384 uid=100

478 xxx  xxx  ACL ENGINE: chk_access exit: type=NFS4 rc=0 ops=16384 priv=0 against=0
478 xxx  xxx  ACL ENGINE: set_acl entry: type=NFS4 ctl_flg=2 obj_mode=33587200 mode=0 size=48

478 xxx  xxx  ACL ENGINE: validate_acl: type=NFS4 rc=22 ace_cnt=1 acl_len=48 size=12
478 xxx  xxx  ACL ENGINE: set_acl exit: type=NFS4 rc=22 obj_mode=33587200 size=68 cmd=536878912
```

A chk_access exit karaktersorozatot tartalmazó második sor jelzi, hogy az ACL írása engedélyezett (rc = 0). A validate_acl karaktersorozatot tartalmazó negyedik és a set_acl exit karaktersorozatot tartalmazó ötödik sor jelzi, hogy az ACL alkalmazása nem volt sikeres (az rc=22 a következőt jelzi: **EINVAL**). A validate_acl karaktersorozatot tartalmazó negyedik sor jelzi, hogy a hozzáférés felügyeleti bejegyzés első sorával probléma van (ace_cnt=1). Az első hozzáférés felügyeleti bejegyzésben - s:(EVERYONE@): a rwxacC) - nincs **p** hozzáférési maszk. A **w** beállítás mellett a **p** beállításra is szükség van az ACL alkalmazásakor.

Hibaelhárítás hozzáférés megtagadva

A fájlrendszer műveletek (például írás vagy olvasás) meghiúsulhatnak a társított NFS4 ACL-lel rendelkező objektumokon. Általában megjelenik egy hibaüzenet, de az üzenet nem tartalmaz elegendő információt a hozzáférési probléma meghatározásához. A hozzáférési probléma megkereséséhez használhatja a nyomkövetés szolgáltatást. Tegyük fel hogy a /j2v2/fájl2 NFS4 ACL-je a következő:

```
s:(EVERYONE@):  a       rwpx
```

A következő parancs az "Engedély megtagadva" üzenetet adja vissza:

```
ls -l /j2v2/file2
```

A probléma hibaelhárításához végezze el az alábbi lépéseket:

1. Futtassa a trace, az ls -l /j2v2/file2 és a trcrpt parancsokat:

```
$ trace -j  478 -o trc.raw
$->!ls -l /j2v2/file2
$ ->quit
$ trcrpt trc.raw > trc.rpt
```

2. Elemezze a nyomkövetési jelentést. A fájl az alábbiakhoz hasonló sorokat tartalmaz:

```
478 xxx  xxx  ACL ENGINE: chk_access entry: type=NFS4 obj_mode=33587711 size=68 ops=1024 uid=100
478  xxx  xxx   ACL ENGINE: nfs4_chk_access_self: type=NFS4 aceN=1 aceCnt=1 req=128 deny=0
478  xxx  xxx   ACL ENGINE: nfs4_mask_privcheck: type=NFS4 deny=128 priv=128
478  xxx  xxx   ACL ENGINE: chk_access exit: type=NFS4 rc=13 ops=1024 priv=0 against=0
```

A harmadik sor jelzi, hogy az access mask = 128 (**0x80**) számára a hozzáférés meg van tagadva, mert az csak READ_ATTRIBUTES jogosultság (lásd: /usr/include/sys/acl.h fájl).

Ellenőrzés áttekintése

Az ellenőrzési alrendszer lehetővé teszi a rendszergazdának hogy feljegyezze a biztonságot illető információkat, amelyek elemezhetőek hogy megtalálja a biztonsági házirend aktuális megsértéseit.

Megfigyelési alrendszer

A megfigyelési rendszer felismerő, adatgyűjtő és feldolgozó funkciókat tartalmaz.

- “Megfigyelési esemény felismerése” oldalszám: 132
- “Eseményinformációk gyűjteménye” oldalszám: 132
- “Megfigyelési napló információk feldolgozása” oldalszám: 132

A rendszeradminisztrátor e funkciók mindegyikét konfigurálhatja.

Megfigyelési esemény felismerése

Az eseményészlelés a Megbízható számítástechnikai alapkörnyezet (TCB) teljes egészére kiterjed, mind a kernelre (felügyelői állapotú kód) és a megbízható programokra (felhasználói állapotú kód). Megfigyelhető esemény a rendszer a biztonsággal kapcsolatos összes eseménye. Biztonsággal kapcsolatos eseménynek számít a rendszer biztonsági állapotának bármilyen változása, a rendszer hozzáférés-vezérlésének vagy az elszámoltathatósági biztonsági irányelveknek a tényleges vagy megkísérelt megsértése. A megfigyelhető eseményeket felismerő programok és kernelmodulok felelősek az események jelentéséért a rendszer megfigyelési naplózó számára, amely a kernel részeként fut és akár szubrutinként (a megbízható program megfigyeléshez), akár kernel eljáráshíváson belül (felügyelői állapotú megfigyeléshez) meghívható. Jelentésre kerül a megfigyelhető esemény neve, az esemény sikeressége vagy sikertelensége, valamint a biztonsági megfigyelésre vonatkozó, az eseménnyel kapcsolatos további információk.

Az eseményfelismerési konfiguráció az eseményfelismerés be- vagy kikapcsolásából áll, valamint annak meghatározásából, hogy mely felhasználókkal kapcsolatban mely események kerüljenek megfigyelésre. Az eseményfelismerés bekapcsolásához, a megfigyelési alrendszer engedélyezéséhez és letiltásához használja az **audit** parancsot. A megfigyelési alrendszer által feldolgozott események és felhasználók az /etc/security/audit/config fájlban találhatók.

Eseményinformációk gyűjteménye

Információgyűjtés alatt a kiválasztott megfigyelhető események naplózását értjük. Ezt a funkciót a kernel megfigyelési naplózója végzi, amely mind rendszerhívási, mind kernelen belüli eljáráshívási felületet biztosít a megfigyelhető események rögzítéséhez.

A megfigyelési naplózó felelős a teljes megfigyelési rekord elkészítéséért, amely két részből áll: a megfigyelési fejléc az összes eseményre egységesen jellemző információkat tartalmazza (ilyen például az esemény neve, a felelős felhasználó, az esemény ideje és visszatérési állapota), a megfigyelési napló pedig az eseményspecifikus információkat. A megfigyelési naplózó minden egyes rekordot sorban a kernel megfigyelési naplóhoz fűz, amelybe kétféle módon (akár egyszerre is) történhet az írás:

BIN mód
A napló váltakozó fájlokba íródik, a megfelelő biztonság és a hosszú távú tárolás érdekében.

STREAM mód
A napló egy körkörös pufferbe íródik, amelyet szinkronizáltan olvas ki egy megfigyelési pszeudoeszköz. A STREAM mód használata esetén azonnali a válasz.

Az információgyűjtés beállítható az előtérben (eseményrögzítés) és a háttérben (naplófeldolgozás). Az eseményrögzítés felhasználónként választható. Minden egyes felhasználóhoz megfigyelési események meghatározott halmaza tartozik, amelyek fellépésük esetén rögzítésre kerülnek a megfigyelési naplóban. A háttérben a módon egyenként állíthatók, vagyis az adminisztrátor kiválaszthatja az adott környezetnek legjobban megfelelő feldolgozási módot. Ezen felül BIN módú megfigyelés esetén beállítható, hogy riasztás történjen, ha a fájlrendszeren a napló számára rendelkezésre álló hely kezd túlságosan kevés lenni.

Megfigyelési napló információk feldolgozása

Az operációs rendszer számos lehetőséget kínál a kernel megfigyelési napló feldolgozására. A BIN módú napló a napló archiválása előtt igény szerint tömöríthető, szűrhető és kimenetként formázható is. A tömörítés Huffman-kódolással történik. A szűrés a szabványos lekérdezőnyelvhez (SQL) hasonló megfigyelési rekordkiválasztással történik (az **auditselect** parancs segítségével), amely lehetővé teszi a megfigyelési napló szelektív megjelenítését és szelektív megtartását is. A megfigyelési napló rekordjainak formázásával megvizsgálható a megfigyelési napló, időszakos biztonsági jelentések készíthetők, illetve papíron is kinyomtatható a megfigyelési napló.

A STREAM módú megfigyelési napló valós időben figyelhető, vagyis azonnali lehet a reagálás a veszélyekre. Mindezen lehetőségek beállítása külön programokkal történik, amelyek démonfolyamatokként hívhatók meg akár a BIN, akár a STREAM módú naplók szűrésére, bár egyes szűrőprogramok természetüknél fogva jobban illeszkednek az egyik módhoz, mint a másikhoz.

Megfigyelési alrendszer konfigurációja

A megfigyelési alrendszerhez tartozik egy globális állapotú változó, amely jelzi, hogy a megfigyelési alrendszer be van-e kapcsolva. Ezen felül minden egyes folyamathoz tartozik egy helyi állapotú változó, amely jelzi, hogy a megfigyelési alrendszer rögzítsen-e információkat az adott folyamatról.

A két változó együttese határozza meg, hogy a Megbízható számítástechnikai alapkörnyezet (TCB) moduljai és programjai milyen eseményeket ismerjenek fel. A TCB megfigyelés egy adott folyamatot illető kikapcsolása nem a rendszer elszámoltathatósági irányelvének megkerülésére szolgál, hanem arra, hogy egy folyamat végezhessen saját megfigyelést. Engedélyezve a megbízható programok számára saját maguk megfigyelését, hatékonyabb információgyűjtés alakítható ki.

Megfigyelési alrendszer-információk gyűjteménye

Az információgyűjtés az eseménykiválasztás és kernel megfigyelési napló módokkal foglalkozik. Ezt a feladatot egy kernel-szubrutin végzi, amely megfelelő felületet kínál a megfigyelhető eseményeket felismerő TCB összetevők által használt naplóinformációk eléréséhez, illetve konfigurációs felületeket, amelyek segítségével a megfigyelési alrendszer vezérelheti a naplózási rutint.

Megfigyelésnaplózás

A megfigyelhető eseményeket két illesztő naplózza: a felhasználói állapot és a felügyelői állapot. A TCB felhasználói állapotú része az **auditlog** vagy **auditwrite** szubrutin használja, míg a TCB felügyelői állapotú része egy sor kerneleljárást hív meg.

Minden egyes rekord esetében a megfigyelési esemény naplózó egy megfigyelési fejlécet helyez az eseményspecifikus információk elé. Ez a fejléc azonosítja a felhasználót és folyamatot, amelyre vonatkozóan az esemény megfigyelésre kerül, valamint rögzíti az esemény időpontját. Az eseményt felismerő kód biztosítja az esemény típusát, valamint visszatérési kódját vagy állapotát, illetve opcionálisan, további eseményspecifikus információkat (eseménynapló). Ilyen eseményspecifikus információk lehetnek például objektumok nevei (például azon fájlok, amelyek visszautasították a hozzáférési kérést, vagy a sikertelen bejelentkezés során használ terminálok), szubrutin-paraméterek, illetve egyéb módosított adatok.

Az események meghatározása szimbolikusan történik, nem pedig numerikusan. Ez csökkenti a névütközések esélyét, akkor is, ha nincs névregisztrációs séma használatban. Mivel a szubrutinok megfigyelhetők, és mivel a kerneldefinícióban nincsenek rögzített kapcsolt virtuális áramkör (SVC) számok, nehéz lenne az eseményeket szám alapján rögzíteni. A kernelillesztő minden egyes bővítésénél vagy újradefiniálásánál át kellene tekinteni és naplózni kellene a számok leképezését.

Megfigyelési rekord formátuma

A megfigyelési rekordok egy egységes fejrészből, valamint az adott rekord megfigyelési eseményére vonatkozó naplórészből állnak. A fejlécek szerkezetét az /usr/include/sys/audit.h fájl határozza meg. A megfigyelési naplókban található információk formátuma az egyes alapeseményeknek megfelelő, és az /etc/security/audit/events fájlban tekinthető meg.

A megfigyelési fejléc információit általában a naplózó rutin gyűjti be, a pontosság biztosítása érdekében, míg a megfigyelési naplókat az a kód szolgáltatja, amelyik ténylegesen felderíti az eseményt. A megfigyelési naplózó nem ismeri a megfigyelési naplók szerkezetét vagy struktúráját. Ha a **login** parancs például felismer egy hibás bejelentkezést, akkor feljegyzi az adott eseményt és hogy melyik terminálon történt, majd a megfigyelési naplóba írja az **auditlog** szubrutin segítségével. A megfigyelési naplózó kernel összetevője feljegyzi a tárgyspecifikus információkat (felhasználói azonosítók, folyamatazonosítók, időpont) egy fejlécbe, majd a többi információhoz

csatolja. A hívónak csak az esemény nevét és a fejléc eredménymezőit kell megadnia.

Megfigyelésnaplózó-konfiguráció

A teljes megfigyelési rekord elkészítéséért a megfigyelési naplózó felelős. Meg kell határoznia, mely megfigyelési események kerüljenek naplózásra.

Megfigyelési események kiválasztása

A megfigyelési események kiválasztásakor az alábbi típusok közül lehet választani:

Folyamatonkénti megfigyelés

Az egyes folyamatok eseményeinek hatékony megfigyelése érdekében az a rendszeradminisztrátor megfigyelési osztályokat definiálhat. A megfigyelési osztály a rendszer alap megfigyelhető eseményeinek egy részhalmaza. A megfigyelési osztályok az alap megfigyelési események kényelmes logikai csoportosítását biztosítják.

A rendszer minden egyes felhasználója számára a rendszeradminisztrátor megadhat egy sor megfigyelési osztályt, amelyek meghatározzák, mely alapesemények kerüljenek rögzítésre az adott felhasználóval kapcsolatban. A felhasználó által futtatott folyamatok felcímkéződnek a megfigyelési osztályokkal.

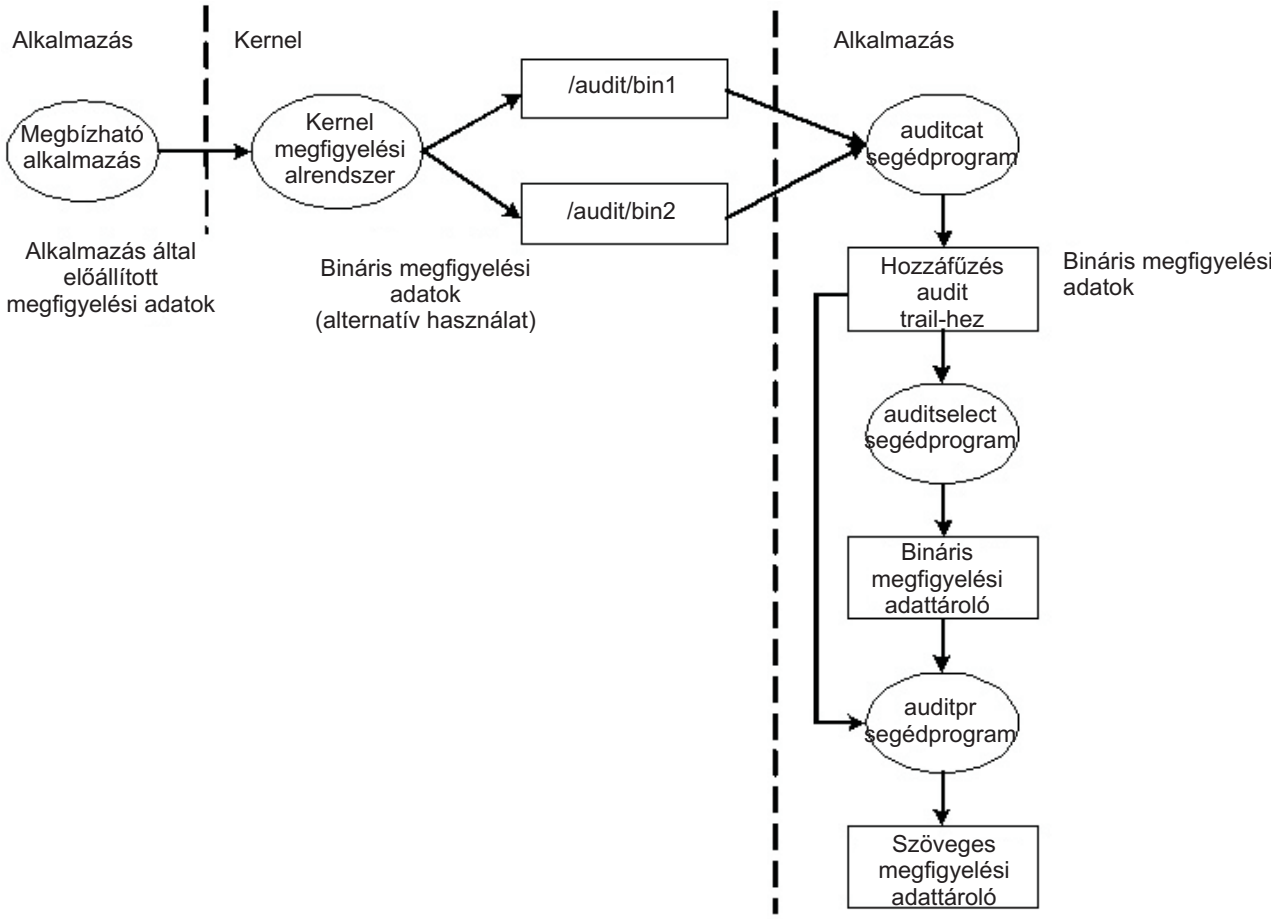
Objektumonkénti megfigyelés

Az operációs rendszer lehetővé teszi a név alapján történő objektum-hozzáférések megfigyelését; más szavakkal, az egyes objektumok (jellemzően fájlok) megfigyelését. A név szerinti objektummegfigyelés révén elkerülhető az összes objektum figyelése csak azért, mert néhány objektum figyelésére szükség van. Ezenfelül megadható a megfigyelés módja, vagyis csak a kívánt módú (írás/olvasás/végrehajtás) hozzáférések kerülnek rögzítésre.

Kernel megfigyelési napló módok

A kernelnaplózás BIN vagy STREAM módjai határozzák meg, hová is íródik ténylegesen a kernel megfigyelési napló. BIN mód használata esetén a kernel megfigyelési naplózónak (a megfigyelés indítása előtt) meg kell adni legalább egy fájlleírót, ahová a rekordokat fűzheti.

BIN módban a megfigyelési rekordok váltakozó fájlokba íródnak. A megfigyelés indulásakor a kernel kap két fájlleírót, valamint egy javasolt maximális fájlméretet. Felfüggeszti a hívó folyamatot, majd megkezdi a megfigyelési rekordok beírását az első fájlleíróba. Ha az első gyűjtő mérete eléri a megadott maximális méretet, valamint ha a megadott második fájlleíró is érvényes, akkor a kernel átkapcsol a második gyűjtőre, majd újra aktiválja a hívó folyamatot. A kernel ezután a második gyűjtőbe ír, egészen addig, amíg egy új érvényes fájlleíróval meg nem hívják. Ha ekkor a második gyűjtő tele van, akkor visszakapcsol az első gyűjtőre, majd azonnal visszatér a hívó folyamathoz. Egyébként a hívó folyamat felfüggesztésre kerül, és a kernel egészen addig ír a második gyűjtőbe, amíg az meg nem telik. A feldolgozás ilyen módon folytatódik, egészen addig, amíg a megfigyelés kikapcsolásra nem kerül. A BIN mód illusztrálására álljon itt egy ábra:

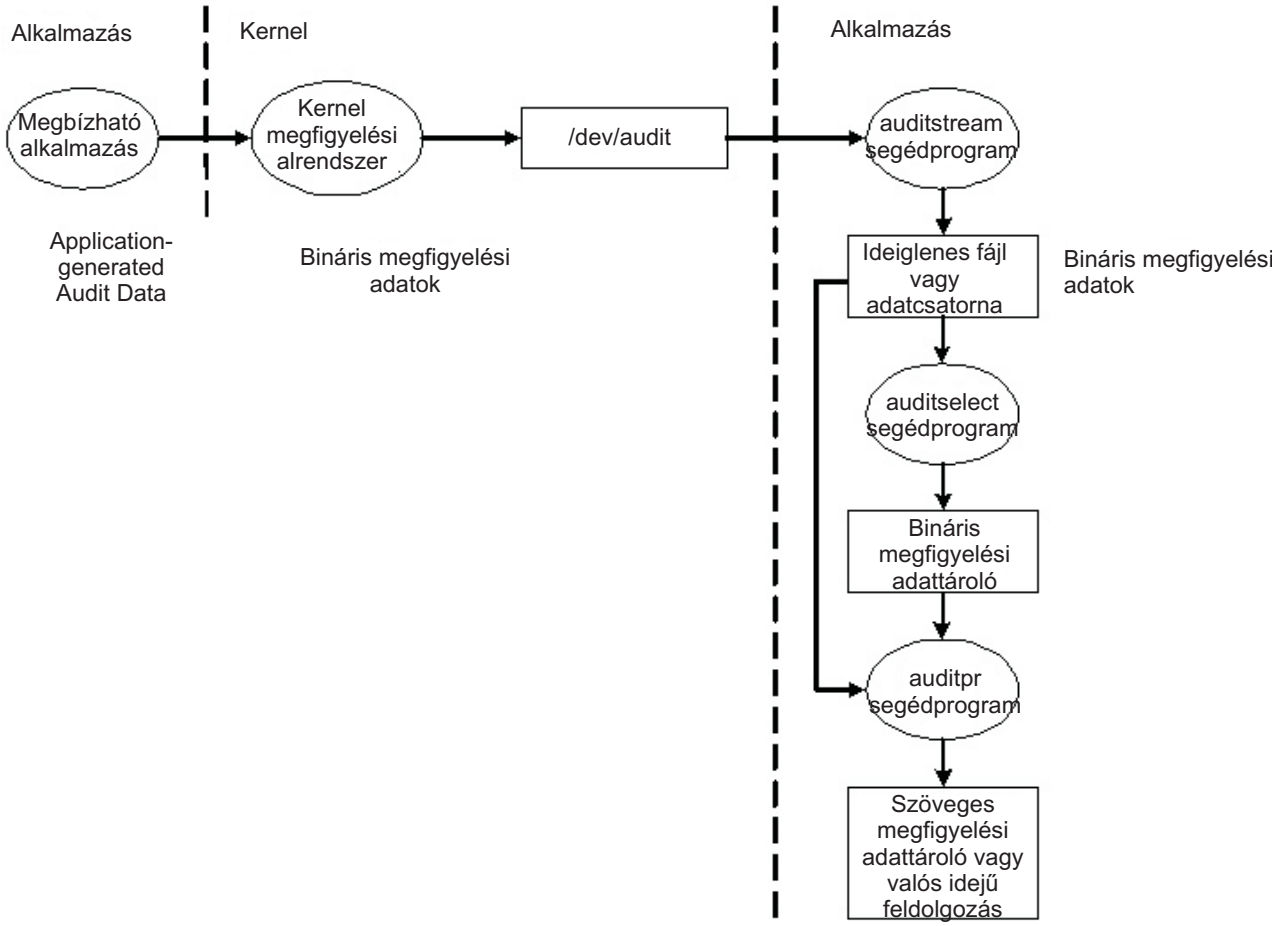


1. ábra: A megfigyelés BIN módjának folyamata.. Ez az ábra a megfigyelés BIN módjának folyamatát mutatja.

A váltakozó gyűjtők mechanizmusával biztosítható, hogy a megfigyelési alrendszernek mindig legyen kiírnivalója a megfigyelési rekordok feldolgozásakor. Amikor a megfigyelési alrendszer átkapcsol a másik gyűjtőre, akkor az első tartalmát a trace fájlba üríti. Így ha újra vissza kell kapcsolni az első gyűjtőre, annak tartalma rendelkezésre áll. Ez a megoldás szétválasztja az adatok tárolását és elemzését azok létrehozásától. Általában a kernel által éppen nem írt megfigyelési gyűjtő kiolvasására az **auditcat** program használatos. Annak biztosítása érdekében, hogy a rendszeren soha ne fogyjon el a megfigyelési napló számára fenntartott terület (az **auditcat** program kimenete), a /etc/security/audit/config fájlban megadható a *freespace* paraméter. Ha a rendszerben az itt megadott számúnál kevesebb 512 bájtos blokk marad, akkor egy syslog üzenet keletkezik.

Ha a megfigyelés engedélyezve van, akkor a /etc/security/audit/config fájl start szakaszában lévő *binmode* paramétert panic értékre kell állítani. A *bin* szakasz freespace paraméterét legalább a megfigyelési naplók tárolására szánt lemezterület 25 százalékára kell állítani. A *bytethreshold* és a *binsize* paramétert 65536 byte-ra kell állítani.

STREAM módban a kernel a rekordokat egy körkörös pufferbe írja. Amikor a kernel eléri a puffer végét, kezdi újra az elején. A folyamatok az adatokat egy /dev/audit nevű pszeudoeszközön keresztül olvashatják ki. Amikor egy folyamat megnyitja ezt az eszközt, létrejön egy csatorna az adott folyamat számára. Opcionálisan, a csatornán beolvasandó események megadhatók, mint megfigyelési osztályok listája. A STREAM megfigyelési módot az alábbi ábra illusztrálja:



2. ábra: A megfigyelés STREAM módjának folyamata. Ez az ábra a megfigyelés STREAM módjának folyamatát mutatja.

A STREAM mód elsődleges célja a megfigyelési napló pontos kiolvasása, amely igen kívánatos valós idejű fenyegetés-figyelés esetén. Egy másik lehetséges használata egy azonnal kiíródó napló létrehozása, vagyis a megfigyelési napló bármilyen megbolygatásának megakadályozása (amely más, írható médiákon tárolt naplók esetén nem előfordulhat).

A STREAM mód ismét egy másik felhasználási módja a megfigyelési adatfolyam egy programba írása, amely eltárolja a megfigyelési információkat egy távoli rendszeren. Ez a megoldás, miközben majdnem valós idejű feldolgozást tesz lehetővé, megakadályozza a megfigyelési információk megbolygatását az eredeti hoszton.

Megfigyelési rekordok feldolgozása

A BIN és STREAM módú megfigyelési rekordok feldolgozására az **auditselect**, az **auditpr** és az **auditmerge** parancsok állnak rendelkezésre. Mindkét segédprogram szűrőként működik, vagyis egyszerűen használhatók csővezetékeken, ami különösen hasznos STREAM módú megfigyelés esetén.

auditselect

Meghatározott megfigyelési rekordok SQL-szerű utasításokkal kiválasztására szolgál. Ha például csak az *afx* felhasználó által előállított **exec()** eseményeket kívánja kiválasztani, akkor írja be a következő parancsot:

```
auditselect -e "login==afx && event==PROC_Execute"
```

auditpr

A bináris megfigyelési rekordokat ember által olvasható formátumra konvertálja. A megjelenített információ mennyisége függ a parancssorban megadott kapcsolóktól. Az **auditpr** által biztosított összes információ az alábbi paranccsal íratható ki:

```
auditpr -v -hhelrtRpPTc
```

A **-v** kapcsoló megadásakor a kernel által minden egyes eseményhez biztosított szokásos megfigyelési információkon kívül a megfigyelési napló (egy eseményspecifikus karaktersorozat) is megjelenítésre kerül (lásd a /etc/security/audit/events fájlt).

auditmerge

Bináris megfigyelési naplók összefésülésére szolgál. Ez különösen akkor hasznos, ha több különböző rendszerről származó megfigyelési naplót kell egyesíteni. Az **auditmerge** parancs fogja a naplók a parancssorban megadott neveit, majd az összefésült bináris naplót a szabványos kimenetre írja ki. Az **auditpr**-re tehát továbbra is szükség van, ha olvasható formában kell kiírni az eredményt. Az **auditmerge** és **auditptr** parancs például így futtatható együtt:

```
auditmerge trail.system1 trail.system2 | auditpr -v -hhelrRtpc
```

Megfigyelési alrendszer használata gyors biztonsági ellenőrzéshez:

Ha csak egyetlen gyanús programot kíván megfigyelni a teljes megfigyelési alrendszer beüzemelése nélkül, akkor használhatja a **watch** parancsot. Ez feljegyzi a megadott program kért, vagy akár összes eseményét.

Ha például meg kívánja tekinteni a **vi /etc/hosts** parancs futtatása során lezajló összes **FILE_Open** eseményt, akkor írja be a következő parancsot:

```
watch -eFILE_Open -o /tmp/vi.watch vi /etc/hosts
```

A /tmp/vi.watch fájl tartalmazni fogja a szerkesztési menet összes **FILE_Open** eseményét.

Eseménykiválasztás

Az eseménykiválasztásnak fenn kell tartania az egyensúlyt a nem elegendő és a túl sok részlet között.

A rendszer megfigyelhető eseményeinek halmaza meghatározza, hogy ténylegesen milyen események figyelhetők meg, illetve a megfigyelés milyen finomsággal történjen. A megfigyelhető eseményeknek ki kell terjedniük a rendszer biztonsággal kapcsolatos eseményeire, az előzőekben meghatározott módon. A megfigyelhető események részletességének mértéke ésszerű egyensúly kell, hogy legyen az elégtelen részletesség - amikor is nehéz az adminisztrátornak megértenie a kiválasztott információkat - és a túlzott részletesség között. Az események meghatározása kihasználja az észlelt események közötti hasonlóságot. Az továbbiakban *felismert eseménynek* tekintjük a megfigyelhető események bármely példányát; tehát egy adott esemény több helyen is előfordulhat. E mögött az a szemlélet áll, hogy a hasonló biztonsági tulajdonságokkal rendelkező felismert események egyazon megfigyelhető eseményként kiválaszthatók legyenek. Az alábbi lista a biztonsági irányelvek eseményeinek osztályozását mutatja be:

- Tárgyi események
  - Folyamat létrehozása
  - Folyamat törlése
  - Tárgy biztonsági jellemzőinek beállítása: felhasználói azonosítók, csoportazonosítók
  - Folyamatcsoport, vezérlő terminál
- Objektumesemények
  - Objektum létrehozása
  - Objektum törlése
  - Objektum megnyitása (beleértve a folyamatokat is, mint objektumokat)
  - Objektum lezárása (beleértve a folyamatokat is, mint objektumokat)
  - Objektum biztonsági jellemzőinek beállítása: tulajdonos, csoport, hozzáférés felügyeleti lista
- Import/Export események

  - Objektum importálása vagy exportálása
- Elszámoltathatósági események
  - Felhasználó felvétele, vagy a felhasználó jellemzőinek módosítása a jelszóadatbázisban
  - Csoport felvétele, vagy a csoport jellemzőinek módosítása a csoportadatbázisban
  - Felhasználói bejelentkezés
  - Felhasználói kijelentkezés
  - Felhasználó hitelesítési információinak módosítása
  - Megbízható útvonalú terminál konfigurációja
  - Hitelesítés konfigurációja
  - Megfigyelési adminisztráció: események és megfigyelési naplók kiválasztása, be- vagy kikapcsolása, felhasználói megfigyelési osztályok definiálása
- Általános rendszeradminisztrációs események
  - Jogosultságok használata
  - Fájlrendszer beállítása
  - Eszközmeghatározások és konfiguráció
  - Rendszerkonfigurációs paraméterek meghatározása
  - Normális rendszer IPL és leállítás
  - RAS konfiguráció
  - Egyéb rendszerkonfiguráció
  - Megfigyelési alrendszer elindítása
  - Megfigyelési alrendszer leállítása
  - Megfigyelési alrendszer lekérdezése
  - Megfigyelési alrendszer alaphelyzetbe állítása
- Biztonsági rendszer megsértései (potenciális)
  - Hozzáférési jogosultságok visszautasításai
  - Jogosultsági problémák
  - Diagnosztika által felderített hibák és rendszerhibák
  - A TCB módosításának kísérlete

Megfigyelési események:

Egy *megfigyelési esemény* a rendszer bármely biztonsággal kapcsolatos eseménye. Biztonsággal kapcsolatos esemény lehet a rendszer biztonsági állapotának megváltozása, a rendszer hozzáférés-vezérlésének vagy az elszámoltathatósági biztonsági irányelveknek a tényleges vagy megkísérelt megsértése. A megfigyelési eseményeket felismerő programok és kernelmodulok jelentik ezeket az események a rendszer megfigyelési naplózó számára, amely a kernel részeként fut, és egy szubrutin használatával (a megbízható program megfigyeléshez), vagy egy kernel eljáráshíváson belül (felügyelői állapotú megfigyeléshez) érhető el. A megfigyelési eseményekben jelentett információk tartalmazzák az esemény nevét, az esemény sikerességét vagy sikertelenségét, valamint a biztonsági megfigyelésre vonatkozó, az eseménnyel kapcsolatos bármilyen további információkat.

Egy tevékenység megfigyeléséhez azonosítani kell a megfigyelési eseményt kezdeményező parancsot vagy folyamatot, és fel kell venni az eseményt az /etc/security/audit/events fájlba. A megfigyelési események a felhasználókhoz rendelése leegyszerűsíthető a hasonló események megfigyelési osztályokba sorolásával. Ezek a megfigyelési osztályok a /etc/security/audit/config fájl classes szakaszában vannak megadva.

A következő táblázat felsorol néhány általános használt megfigyelési eseményt, amelyek az AIX operációs rendszeren előfordulhatnak:

11. táblázat: Megfigyelési események

Felhasználói vagy rendszer hívás	Megfigyelési esemény	Leírás
fork	PROC_Create	Megadja, hogy egy folyamat létrehozásra kerül.
exit	PROC_Delete	Megadja, hogy a hívó folyamat befejeződött.
exec	PROC_Execute	Futtat egy új programot.
setuidx	PROC_RealUID	Beállítja a folyamat felhasználói azonosítóját.
	PROC_AuditID	
	PROC_SetUserIDs	
setgidx	PROC_RealGID	Beállítja a folyamat csoportazonosítóját.
accessx	FILE_Accessx	Megállapítja egy fájl hozzáférhetőségét.
statacl	FILE_StatAcl	Lekéri egy fájl hozzáférés-felügyeleti információit.
revoke	FILE_Revoke	Visszahívja egy fájl hozzáférését minden folyamat esetén.
frevoke	FILE_Frevoke	Visszahívja egy fájl más folyamatok általi hozzáférését.
usrinfo	PROC_Environ	Módosítja a felhasználói információ adatok egy részét.
sigaction	PROC_SetSignal	Megadja a műveletet, amelyet végre kell hajtani, amikor egy jelzés érkezik a szubrutint kibocsátó folyamathoz.
setrlimit	PROC_Limits	A maximális rendszer erőforrások fogyasztását vezérli.
nice	PROC_SetPri	Megadja a nice függvény használatát.
setpri	PROC_Setpri	beállítja a folyamatok rögzített prioritását.
setpriv	PROC_Privilege	Módosítja a folyamatokhoz tartozó vektorok jogosultságait.
settimer	PROC_Settimer	Beállítja egy megadott rendszer szintű időmérő aktuális értékét.
adjtime	PROC_Adjtime	Módosítja a rendszerórát.
ptrace	PROC_Debug	Nyomon követi egy másik folyamat végrehajtását.
kill	PROC_Kill	Elküld egy jelzést egy folyamatnak vagy a folyamatok egy csoportjának.
setpgid	PROC_setpgid	Beállítja a folyamat csoportazonosítóját.
ld_loadmodule	PROC_Load	Betölt egy új objektummodult a folyamat címtérbe.
	PROC_LoadError	Jelzi, hogy az objektumbetöltés meghiúsult.
setgroups	PROC_SetGroups	Módosítja a folyamat párhuzamos csoportkészletét.
sysconfig	PROC_Sysconfig	Rögzíti a kernel- vagy rendszerkonfiguráción elvégzett műveletet.
audit	AUD_It	Elindítja vagy leállítja a megfigyelési műveletet. Továbbá lekérdezi a megfigyelés állapotát.
auditbin	AUD_Bin_Def	Módosítja az auditbin rendszerhívást.
auditevents	AUD_Events	Eseményeket módosít.
auditobj	AUD_Objects	Módosítja az auditobj rendszerhívást.
auditproc	AUD_Proc	Lekérdezi vagy beállítja egy folyamat megfigyelési állapotát.
acct	ACCT_Disable	Letiltja a rendszer elszámolást.
	ACCT_Enable	Engedélyezi a rendszer elszámolást.

11. táblázat: Megfigyelési események (Folytatás)

Felhasználói vagy rendszer hívás	Megfigyelési esemény	Leírás
open és create	FILE_Open	Meghívja az **open** szubrutint.
read	FILE_Read	Adatokat olvas be a fájlleíróból.
write	FILE_Write	Adatokat ír a fájlleíróba.
close	FILE_Close	Bezárja a megnyitott fájlleírót.
link	FILE_Link	Létrehoz egy új könyvtárbejegyzést egy fájlrendszer objektumhoz.
unlink	FILE_Unlink	Eltávolít egy fájlrendszer objektumot.
rename	FILE_Rename	Módosítja egy fájlrendszer objektum nevét.
chown	FILE_Owner	Módosítja a fájl tulajdonjogát.
chmod	FILE_Mode	Módosítja a fájlmódot.
fchmod	FILE_Fchmod	Módosítja egy fájlleíró fájlengedélyeit.
fchown	FILE_Fchown	Módosítja egy fájlleíró tulajdonjogát.
truncate	FILE_Truncate	Módosítja normál fájlok vagy egy megosztott memória objektum hosszát.
symlink	FILE_Symlink	Létrehoz egy szimbolikus hivatkozást.
pipe	FILE_Pipe	Létrehoz egy névtelen adatcsatornát.
mknod	FILE_Mknod	Létrehoz egy speciális eszközfájlt vagy egy első-be-első-ki (first-in-first-out - FIFO) speciális fájlt.
fcntl	FILE_Dupfd	Többszörözi a fájlleírót.
fscntl	FS_Extend	Kiterjeszti a fájlrendszert.
mount	FS_Mount	Csatlakoztatja a fájlrendszert egy megnevezett könyvtárhoz.
umount	FS_Umount	Leválasztja a beillesztett fájlrendszert.
chacl	FILE_Acl	Módosítja egy fájl hozzáférés-felügyeleti listáját (ACL).
fchacl	FILE_Facl	Módosítja egy fájlleíró ACL listáját.
chpriv	FILE_Privilege	Beállítja egy fájlútvonalnév Jogosultságvezérlő listáját (PCL).
	FILE_Chpriv	Módosítja a PCL listát.
	FILE_Fchpriv	Módosítja egy fájlleíró PCL listáját.
chdir	FS_Chdir	Átváltja a jelenlegi munkakönyvtárat.
fchdir	FS_Fchdir	Módosítja a jelenlegi munkakönyvtárat egy fájlleíró használatával.
chroot	FS_Chroot	Módosítja a gyökérkönyvtár (/) jelentését az aktuális folyamat esetén.
rmdir	FS_Rmdir	Eltávolítja a könyvtár objektumot.
mkdir	FS_Mkdir	Létrehoz egy könyvtárat.
utimes	FILE_Utimes	Meghívja az **utimes** szubrutint.
stat	FILE_Stat	Meghívja a **stat** szubrutint.
msgget	MSG_Create	Létrehoz egy üzenetsort.
msgrcv	MSG_Read	Üzenetet fogad egy üzenetsorból.
msgsnd	MSG_Write	Üzenetet küld egy üzenetsorba.
msgctl	MSG_Delete	Eltávolít egy üzenetsort.
	MSG_Owner	Changes ownership and access right of a message queue.
	MSG_Mode	Lekérdezi egy üzenetsor hozzáférési jogosultságait.

11. táblázat: Megfigyelési események (Folytatás)

Felhasználói vagy rendszer hívás	Megfigyelési esemény	Leírás
semget	SEM_Create	Létrehoz egy szemaforkészletet.
semop	SEM_Op	Növel vagy csökkent egy vagy több szemafort.
semctl	SEM_Delete	Töröl egy szemaforkészletet.
	SEM_Owner	Módosítja egy szemaforkészlet tulajdonjogát és hozzáférési jogosultságait.
	SEM_Mode	Lekérdezi egy szemaforkészlet hozzáférési jogosultságait.
shmget	SHM_Create	Létrehoz egy új megosztott memóriaszegmenst.
shmat	SHM_Open	Meghívja a **shmat** szubrutint az **Open** paraméterrel.
shmat	SHM_Detach	Meghívja a **shmat** szubrutint a **Detach** paraméterrel.
shmctl	SHM_Close	Bezárja a megosztott memóriaszegmenst.
	SHM_Owner	Módosítja egy megosztott memóriaszegmens tulajdonjogát és hozzáférési jogosultságait.
	SHM_Mode	Lekérdez egy megosztott memóriaszegmens hozzáférési jogosultságait.
tcpip user level	TCPIP_config	Naplózza a TCP/IP felület módosításait.
	TCPIP_host_id	Naplózza a rendszer hosztnév módosítására tett kísérleteket.
	TCPIP_route	Naplózza az útvonalkezelési tábla módosításait.
	TCPIP_connect	Meghívja a **connect** szubrutint.
	TCPIP_data_out	Adatok elküldve.
	TCPIP_data_in	Adatok fogadva.
	TCPIP_set_time	Naplózza a rendszeridő hálózaton keresztüli módosítására tett kísérleteket.
tcpip kernel level	TCP_ksocket	Meghívja a kernel TCP/IP kernel szolgáltatását.
	TCP_ksocketpair	
	TCP_kclose	
	TCP_ksetopt	
	TCP_kbind	
	TCP_klisten	
	TCP_kconnect	
	TCP_kaccept	
	TCP_kshutdown	
	TCP_ksend	
	TCP_kreceive	
tsm	USER_Login	Bejelentkezteti a felhasználót a rendszerbe.
	PORT_Locked	Jelzi, hogy a port zárolva van érvénytelen bejelentkezési kísérletek miatt.
	TERM_Logout	Kijelentkezteti a felhasználót a rendszerből.
rlogind vagy telnetd	USER_Exit	Jelzi, hogy a felhasználó ki van jelentkezve.
usrck	USER_Check	Ellenőrzi egy felhasználómeghatározás pontosságát.
	USRCK_Error	
logout	USER_Logout	Leállítja az összes folyamatot egy porton.
chsec	PORT_Change	Jelzi a port attribútumértékek változását.
chuser	USER_Change	Felhasználó attribútumokat módosít.
rmuser	USER_Remove	Eltávolít egy felhasználót.

11. táblázat: Megfigyelési események (Folytatás)

Felhasználói vagy rendszer hívás	Megfigyelési esemény	Leírás
mkuser	USER_Create	Létrehoz egy felhasználót.
setgroups	USER_SetGroups	Beállítja a jelenlegi folyamat kiegészítő csoportazonosítóját.
setsenv	USER_SetEnv	Beállítja a környezeti változót.
su	USER_SU	Módosítja a munkamenettel társított felhasználói azonosítót.
grpck	GROUP_User	Eltávolít nem létező felhasználókat a csoportból.
	GROUP_Adms	Eltávolít nem létező adminisztrátori felhasználókat a csoportból.
chgroup	GROUP_Change	Csoport attribútumokat módosít.
mkgroup	GROUP_Create	Létrehoz egy csoportot.
rmgroup	GROUP_Remove	Eltávolít egy csoportot.
passwd	PASSWORD_Change	Módosít egy felhasználói jelszót.
pwdadm	PASSWORD_Flags	Módosít egy adminisztrátori jelszót.
pwdck	PASSWORD_Check	Ellenőrzi a helyi hitelesítési információk pontosságát.
	PASWORD_Ckerr	
startsrc	SRC_Start	Elindít egy rendszer erőforrás vezérlőt.
stopsrc	SRC_Stop	Leállít egy rendszer erőforrás vezérlőt.
addssys	SRC_Addssys	Hozzáadja a SRCsubsys meghatározást az alrendszer objektumosztályhoz.
chssys	SRC_Chssys	Módosít egy alrendszer-meghatározást az alrendszer objektumosztályban.
addserver	SRC_Addserver	Hozzáad egy alkiszolgáló-meghatározás az alkiszolgáló objektumosztályhoz.
chserver	SRC_Chserver	Módosít egy alkiszolgáló-meghatározás az alkiszolgáló objektumosztályban.
rmsys	SRC_Delssys	Eltávolít egy alrendszer-meghatározást az alrendszer objektumosztályból.
rmserver	SRC_Delserver	Eltávolít egy alkiszolgáló-meghatározást a Subserver típus objektumosztályból.
enq	ENQUE_admin	Berak a sorba egy fájlt.
qdaemon	ENQUE_exec	Sorba állított munkákat ütemez.
sendmail	SENDMAIL_Config	továbbítja a levelet a helyi vagy hálózati kézbesítéshez.
	SENDMAIL_ToFile	
at	AT_JobAdd	Eltávolítja vagy hozzáadja a parancsokat, amelyek futtatása az **at** parancs használatával vannak ütemezve.
	At_JobRemove	
cron	CRON_JobRemove	Eltávolítja vagy hozzáadja a parancsokat, amelyek futtatása a **cron** parancs használatával vannak ütemezve.
	CRON_JobAdd	
	CRON_Start	Jelzi egy **cron** munka kezdetét.
	CRON_Finish	Jelzi egy **cron** munka végét.
nvload	NVRAM_Config	Megadja a hozzáférést a nem felejtő kötetlen elérésű memóriához (NVRAM).
cfgmgr	DEV_Configure	Eszközöket konfigurál.
chdev and mkdev	DEV_Change	Megad egy módosítást az eszközben.
mkdev	DEV_Create	Megadja, hogy az eszköz létre van hozva.
	DEV_Start	Megadja, hogy az eszköz el van indítva.

11. táblázat: Megfigyelési események (Folytatás)

Felhasználói vagy rendszer hívás	Megfigyelési esemény	Leírás
installp	INSTALLP_Inst	Telepíti az elérhető szoftvertermékeket egy kompatibilis telepítési csomagba.
	INSTALLP_Exec	
rmdev	DEV_Stop	Megadja, hogy az eszköz le van állítva.
	DEV_Unconfigure	Megadja, hogy az eszköz nincs konfigurálva.
	DEV_Remove	Megadja, hogy az eszköz eltávolításra került.
lchangelv, lextendlv, and lreducelv	LVM_ChangeLV	Megadja, hogy a logikai kötetek módosítva lettek.
lchangepv, ldeletepv, and linstallpv	LVM_ChangeVG	Megadja, hogy a kötetcsoport módosítva lett.
lcreatelv	LVM_CreateLV	Megadja, hogy egy logikai kötet hozzáadásra került a rendszerhez.
lcreatevg	LVM_CreateVG	Megadja, hogy egy kötetcsoport létrehozásra került a rendszerben.
ldeletepv	LVM_DeleteVG	Megadja, hogy a kötetcsoport eltávolításra került a rendszerből.
rmlv	LVM_DeleteLV	Megadja, hogy a logikai kötet eltávolításra került a rendszerből.
lvaryoffvg	LVM_VaryoffVG	Deaktivál egy kötetcsoportot.
lvaryonvg	LVM_VaryonVG	Aktivál egy kötetcsoportot.
Logikai kötet műveletek	LVM_AddLV	Hozzáad egy logikai kötetet egy meglévő kötetcsoporthoz.
	LVM_KDeleteLV	Eltávolít egy logikai kötetet egy meglévő kötetcsoportból.
	LVM_ExtendLV	Növeli egy logikai kötet méretét a kötetcsoportról leválasztott fizikai partíciók hozzáadásával.
	LVM_ReduceLV	Csökkenti egy logikai kötet méretét.
	LVM_KChangeLV	Módosít egy meglévő logikai kötetet.
	LVM_AvoidLV	Nem engedélyezi, hogy egy logikai kötet adott műveleteket hajtson végre.
Fizikai kötet műveletek	LVM_MissingPV	Hozzáad egy fizikai kötetet egy meglévő kötetcsoporthoz.
	LVM_AddPV	Hozzáad egy fizikai kötetet egy meglévő kötetcsoporthoz
	LVM_AddMissPV	Hozzáad egy hiányzó fizikai kötetet egy meglévő kötetcsoporthoz
	LVM_DeletePV	Töröl egy fizikai kötetet egy meglévő kötetcsoportból.
	LVM_RemovePV	Eltávolít egy fizikai kötetet egy meglévő kötetcsoportból.
	LVM_AddVGSA	Hozzáad egy kötetcsoport állapotterületet (VGSA) egy meglévő fizikai kötethez.
	LVM_DeleteVGSA	Eltávolít egy VGSA területet egy meglévő fizikai kötetből.
Kötetcsoport műveletek	LVM_SetupVG	Beállítja a kötetcsoportot logikai kötetek meghatározásával, illetve a VGSA és tükör írási konzisztencia gyorsítótár (MWCC) információinak megadásával.
	LVM_DefineVG	Meghatározza a kötetcsoportot a kernelhez.
	LVM_KDeleteVG	Töröl egy kötetcsoportot a kernelből.

11. táblázat: Megfigyelési események (Folytatás)

Felhasználói vagy rendszer hívás	Megfigyelési esemény	Leírás
Mentési és visszaállítási műveletek	BACKUP_Export	Rögzíti a biztonsági mentési művelet előrehaladását.
	RESTORE_Import	Rögzíti a visszaállítási művelet előrehaladását.
shell	USER_Shell	Rögzíti a felhasználó tty információit.
reboot	USER_Reboot	Rögzíti a rendszer újraindítás eseményét.
	PROC_Reboot	Rögzíti a folyamat újraindítás eseményét. A **reboot** szubrutin újraindítja a rendszert, vagy megismétli a kezdeti programbetöltési (IPL) műveletet a rendszeren.

Megfigyelés beállítása

Az alábbi eljárás egy megfigyelési alrendszer beállítását mutatja be. Amennyiben részletesebb információkra van szükség, forduljon a lépésekben említett konfigurációs fájlokhoz.

1. Válassza ki a megfigyelni kívánt rendszertevékenységeket (eseményeket) az /etc/security/audit/events fájl listájából. Ha új megfigyelési eseményeket adott hozzá az alkalmazásokhoz vagy kernelkiterjesztésekhez, akkor módosítania kell a fájlt, hogy tartalmazza az új eseményeket.
   - Akkor adhat hozzá egy eseményt ehhez a fájlhoz, ha készített az eseményt naplózó kódot egy alkalmazásprogramban (az **auditwrite** vagy **auditlog** szubrutin segítségével) vagy egy kernelkiterjesztésben (az **audit_svcstart**, **audit_svcbcopy**, és **audit_svcfinis** kernelszolgáltatások segítségével).
   - Ellenőrizze, hogy az új megfigyelési eseményekre vonatkozó formázási utasítások szerepelnek az /etc/security/audit/events fájlban. Ezek a meghatározások engedélyezik az **auditpr** parancs számára, hogy megfigyelési naplót írjon a formázott megfigyelési rekordokból.
2. Csoportosítsa a hasonló megfigyelési eseményeket *megfigyelési osztályokba*. Ezeket a megfigyelési osztályokat a /etc/security/audit/config fájl classes szakaszában kell megadnia.
3. Rendelje a megfigyelési osztályokat az egyes felhasználókhoz, és rendeljen megfigyelési eseményeket a megfigyelni kívánt fájlokhoz (objektumokhoz), az alábbiak szerint:
   - A megfigyelési osztályok egy adott felhasználóhoz rendeléséhez írjon be egy sort az /etc/security/audit/config fájl users szakaszába. A megfigyelési osztályok egy felhasználóhoz rendeléséhez használja a **chuser** parancsot.
   - A megfigyelési események egy objektumhoz (adathoz vagy végrehajtható fájlhoz) rendeléséhez írjon be egy sor az /etc/security/audit/objects fájl az adott fájlra vonatkozó szakaszába.
   - Az /usr/lib/security/mkuser.default fájl módosításával alapértelmezett megfigyelési osztályokat adhat meg az új felhasználók számára. Ez a fájl tartalmazza az új felhasználói azonosítók előállításakor használt felhasználói jellemzőket. Használhatja például a general megfigyelési osztályt minden új felhasználói azonosító esetében, az alábbiak szerint:

     ```
     user:
         auditclasses = general
         pgrp = staff
         groups = staff
         shell = /usr/bin/ksh
         home = /home/$USER
     ```

     Az összes megfigyelési esemény megadásához az ALL osztályt adja meg. Ha így tesz, akkor még viszonylag szerény terhelésű rendszer esetén is rendkívül sok adat fog generálódni. Általában célszerű korlátozni a rögzített események számát.
4. Az /etc/security/audit/config fájlban adja meg az adatgyűjtés módját: BIN típusú gyűjtés, STREAM típusú gyűjtés, vagy mindkettő. Győződjön meg róla, hogy a megfigyelési adatok nem ütköznek más adatokkal; használjon külön fájlrendszert a megfigyelési adatokhoz. Ez biztosítja, hogy a megfigyelési adatoknak legyen elegendő hely. Az alábbiak szerint állítsa be az adatgyűjtés típusát:
   - BIN gyűjtemény beállítása:
     a. A binmode = on beírásával a start szakaszban engedélyezze a BIN típusú gyűjtést.

b. Módosítsa a binmode szakaszt és állítsa be a gyűjtőket és a naplót, majd adja meg a BIN mód háttérfeldolgozó parancsait tartalmazó fájl elérési útját. A háttérfeldolgozó parancsok alapértelmezett fájlja az /etc/security/audit/bincmds fájl.
c. Ellenőrizze, hogy a megfigyelési gyűjtőfájlok elegendően nagyok, és állítsa be a *freespace* paramétert úgy, hogy a fájlrendszer beteléséhez közeledvén figyelmeztetést kapjon.
d. Írja be a megfigyelési gyűjtőket feldolgozó parancsértelmező-parancsokat az /etc/security/audit/bincmds fájl egy megfigyelési csövébe.

- STREAM gyűjtemény beállítása:
  a. A start szakaszban streammode = on beírásával engedélyezze a STREAM módú gyűjtést.
  b. Módosítsa a streammode szakaszt és adja meg az a streammode feldolgozó parancsait tartalmazó fájl elérési útját. Az információkat feldolgozó parancsok alapértelmezett fájlja az /etc/security/audit/streamcmds fájl.
  c. Írja be a folyam rekordjait feldolgozó parancsértelmező-parancsokat az /etc/security/audit/streamcmds fájl egy megfigyelési csövébe.

5. Ha elkészült a konfigurációs fájlok szükséges módosításaival, akkor készen áll a megfigyelési alrendszert indító **audit start** parancs kiadására. Ez létrehozza az **AUD_It** eseményt 1 értékkel.
6. A megfigyelt események és objektumok lekérdezésére használja az **audit query** parancsot. Ez létrehozza az **AUD_It** eseményt 2 értékkel.
7. A megfigyelési alrendszer leállítására használja az **audit shutdown** parancsot. Ez létrehozza az **AUD_It** eseményt 4 értékkel.

Általános megfigyelési napló előállítása:

Néhány példa az általános megfigyelési napló előállítására.

Az alábbi példában tegyük fel, hogy a rendszeradminisztrátor a megfigyelési alrendszert egy nagy, sokfelhasználós szerverrendszer figyelésére kívánja használni. Nincs szükség behatolás-felismerő rendszer integrálására, az összes megfigyelési rekord kézzel lesz megvizsgálva, nem tartalmaz-e rendellenességeket. Csupán néhány alapvető megfigyelési esemény kerül rögzítésre, hogy a létrejövő adatok mennyisége az ésszerű határokon belül maradjon.

Az eseményfelismeréshez használni kívánt megfigyelési események az alábbiak:

FILE_Write
Tudni akarjuk, írja-e valaki a konfigurációs fájlokat, ezért ez az esemény az /etc fa összes fájljára vonatkozni fog.

PROC_SetUserIDs
A felhasználói azonosítók minden módosítása

AUD_Bin_Def
A megfigyelési gyűjtő konfigurálása

USER_SU
Az **su** parancs

PASSWORD_Change
A **passwd** parancs

AUD_Lost_Rec
Értesítés elveszett rekordokról

CRON_JobAdd
új cron feladatok

AT_JobAdd
új at feladatok

USER_Login
Minden bejelentkezés

PORT_Locked
Minden túl sok érvénytelen kísérlet miatti zárolás a terminálokon

Az alábbiakban mintát mutatunk egy általános megfigyelési napló létrehozására:

1. Az objects fájlban készítse el a módosítást illetően figyelendő, kritikus fontosságú fájlok listáját (például az /etc könyvtár összes fájlja) és állítsa be rájuk a **FILE_Write** események figyelését:
   ```
   find /etc -type f | awk '{printf("%s:\n\tw = FILE_Write\n\n",$1)}' >> /etc/security/audit/objects
   ```
2. Az **auditcat** paranccsal állítsa be a BIN módú megfigyelést. A /etc/security/audit/bincmds fájl az alábbihoz hasonló:
   ```
   /usr/sbin/auditcat -p -o $trail $bin
   ```
3. Módosítsa az /etc/security/audit/config fájlt és vegyen fel egy osztályt a figyelendő eseményekhez. Sorolja fel az összes létező felhasználót és rendelje mindegyikükhöz a custom osztályt.
   ```
   start:
           binmode = on
           streammode = off

   bin:
           cmds = /etc/security/audit/bincmds
           trail = /audit/trail
           bin1 = /audit/bin1
           bin2 = /audit/bin2
           binsize = 100000
           freespace = 100000

   classes:
           custom = FILE_Write,PROC_SetUser,AUD_Bin_Def,AUD_Lost_Rec,USER_SU, \
                    PASSWORD_Change,CRON_JobAdd,AT_JobAdd,USER_Login,PORT_Locked

   users:
           root = custom
           afx = custom
           ...
   ```
4. Vegye fel a custom megfigyelési osztályt az /usr/lib/security/mkuser.default fájlba, hogy az új felhasználói azonosítókhoz automatikusan a megfelelő megfigyelési osztály legyen hozzárendelve:
   ```
   user:
       auditclasses = custom
       pgrp = staff
       groups = staff
       shell = /usr/bin/ksh
       home = /home/$USER
   ```
5. Az SMIT-tel vagy a **crfs** paranccsal hozzon létre egy /audit nevű új fájlrendszert. A fájlrendszernek elegendően nagynak kell lennie két gyűjtő és egy nagy megfigyelési napló tárolásához.
6. Futtassa az **audit start** parancsot és vizsgálja meg a /audit fájlt. Kezdetben két gyűjtőfájlt és egy üres trail nevű fájlt kell látnia. Ha már használja a rendszert egy ideje, akkor a megfigyelési rekordokat a trail fájlban kell gyűjteni, amelyek az alábbi paranccsal olvashatók ki:
   ```
   auditpr  -hhelpPRtTc -v | more
   ```

A fenti példában csupán néhány eseményt használtunk. Az összes esemény megtekintéséhez megadható az ALL osztálynév minden felhasználóhoz. Ez nagyon nagy mennyiségű adatot generál. Ha kívánja, a custom osztályt bővítheti a felhasználói és jogosultság módosításokkal kapcsolatos összes eseménnyel.

Kritikus fájlok elérésének megfigyelése valós időben:

Ezen lépések segítségével megfigyelhető a kritikus fájlok hozzáférése valós időben.

Tegye a következőket:

1. Az objects fájlban készítse el a módosítást illetően figyelendő, kritikus fontosságú fájlok listáját (például az /etc könyvtár összes fájlja) és állítsa be rájuk a **FILE_Write** események figyelését:
   ```
   find /etc -type f | awk '{printf("%s:\n\tw = FILE_Write\n\n",$1)}' >> /etc/security/audit/objects
   ```
2. Állítsa be a STREAM típusú megfigyelést úgy, hogy az listázza az összes fájlírást. (Ebben a példában a konzolra listázzuk a fájlírásokat, de éles környezetben praktikusan egy háttérrendszer küldje az eseményeket egy behatolás-felismerő rendszer felé.) Az /etc/security/audit/streamcmds fájl tehát az alábbihoz hasonlóan nézzen ki:
   ```
   /usr/sbin/auditstream | /usr/sbin/auditselect -e "event == FILE_Write" |
   auditpr  -hhelpPRtTc -v > /dev/console &
   ```
3. Állítsa be az /etc/security/audit/config fájlban a STREAM típusú adatgyűjtést, vegyen fel egy osztályt a fájlírás eseményekhez, és konfiguráljon minden olyan felhasználót, akiket ezzel az osztállyal kell megfigyelni:
   ```
   start:
           binmode = off
           streammode = on

   stream:
           cmds = /etc/security/audit/streamcmds

   classes:
           filemon = FILE_write

   users:
           root = filemon
           afx = filemon
           ...
   ```
4. Most futtassa le az **audit start** parancsot. Minden **FILE_Write** esemény kiíródik a konzolra.

Megfigyelési események kiválasztása:

A megfigyelés célja azon tevékenységek felismerése, amelyek veszélyeztethetik a rendszer biztonságát.

Ha jogosulatlan felhasználó végzi, az alábbi tevékenységek sértik a rendszer biztonságát és jó alanyai a megfigyelésnek:

- A Megbízható számítástechnikai alapkörnyezet tevékenységei
- Felhasználók hitelesítése
- Hozzáférés a rendszerhez
- A rendszer konfigurációjának módosítása
- A megfigyelési rendszer megkerülése
- A rendszer inicializálása
- Programok telepítése
- Fiókok módosítása
- Információk átvitele a rendszerbe vagy a rendszerből

A megfigyelési rendszernek nincs alapértelmezett megfigyelendő eseményhalmaza. Az igényeknek megfelelően kell kiválasztani az eseményeket vagy eseményosztályokat.

Egy tevékenység megfigyeléséhez azonosítani kell a megfigyelési eseményt kezdeményező parancsot vagy folyamatot, és fel kell venni az eseményt az /etc/security/audit/events fájlba. Ezután az eseményt fel kell venni vagy az /etc/security/audit/config fájl egy megfelelő osztályába, vagy az /etc/security/audit/objects fájl egy objektumszakaszába. A megfigyelési eseményekkel és a napló formázási utasításaival kapcsolatban tekintse meg a /etc/security/audit/events fájlt. A megfigyelési események formátumainak kiírásával és használatával kapcsolatban tekintse meg az **auditpr** parancsot.

A megfigyelni kívánt események kiválasztása után a hasonló eseményeket célszerű osztályokba csoportosítani. A megfigyelési osztályok ezután a felhasználókhoz rendelhetők.

Megfigyelési osztályok kiválasztása

A megfigyelési események a felhasználókhoz rendelése leegyszerűsíthető a hasonló események megfigyelési osztályokba sorolásával. Ezek a megfigyelési osztályok a /etc/security/audit/config fájl classes szakaszában vannak megadva.

Néhány tipikusan használt megfigyelési osztály lehet például az alábbi:

general
A rendszer állapotát megváltoztató, felhasználói hitelesítést igénylő események. A rendszer hozzáférés-vezérlését megkerülő kísérletek megfigyelése.

objects Biztonsági konfigurációs fájlok írási hozzáférése.

kernel A kernel osztály eseményeit a kernel folyamatkezelő funkciói generálják.

Például így nézhet ki az /etc/security/audit/config fájl egy szakasza:

```
classes:
    general = USER_SU,PASSWORD_Change,FILE_Unlink,FILE_Link,FILE_Rename
    system = USER_Change,GROUP_Change,USER_Create,GROUP_Create
    init = USER_Login,USER_Logout
```

Megfigyelés adatgyűjtési metódusának kiválasztása

Az, hogy milyen adatgyűjtési módszert választ, attól függ, hogyan szándékozik használni a megfigyelési adatokat. Ha sok adatot kíván hosszabb időre megőrizni, válassza a BIN típusú adatgyűjtést. Ha a gyűjtéssel egyidejűleg fel kívánja dolgozni az adatokat, válassza a STREAM típust. Ha mindkettőre szükség van, válassza mindkét típusú gyűjtést. Az egyes módszerek leírása:

BIN típusú gyűjtés
Lehetővé teszi egy nagy megfigyelési napló hosszú idejű tárolását. A megfigyelési rekordok egy ideiglenes gyűjtőként szolgáló fájlba íródnak. A fájl megtelése után az **auditbin** démon dolgozza fel az adatokat; a megfigyelési alrendszer a másik gyűjtőfájlba ír, a rekordok pedig egy ellenőrzési naplóba kerülnek.

STREAM típusú gyűjtés
Lehetővé teszi a megfigyelési adatok azonnali feldolgozását. A megfigyelési rekordok egy körkörös pufferbe íródnak a kernelen belül, és a /dev/audit olvasásával kérhetők le. A megfigyelési rekordok megjeleníthetők, kinyomtathatók egy papír alapú megfigyelési napló készítéséhez, vagy gyűjtőrekordokká konvertálhatók az **auditcat** paranccsal.

Munkapartíció megfigyelése

Háromféle megfigyelési típus áll rendelkezésre WPAR környezetben: globális, rendszer és megfigyelés globálisból.

A megfigyelést engedélyezheti globális WPAR-ben és/vagy egy WPAR-on belül. A rendszer és globális WPAR megfigyeléso beállítása hasonló a nem wpar környezet konfigurációjához. Kezdeményezhet globális WPAR megfigyelést a rendszer és alkalmazás WPAR-ekhez.

Megjegyzés: Az alkalmazás WPAR-ek megfigyelése nem kezdeményezhető WPAR-en belülről, de globális WPAR megfigyelésével igen.

A globális WPAR megfigyelés segít a globális rendszeradminisztrátoroknak a WPAR-ek globális rendszerről történő megfigyelésének elvégzésében. A globális rendszeradminisztrátor szabályozni tudja az egyes WPAR-ek megfigyelési szintjét egyetlen helyről a megfigyelendő osztályok megadásával az egyes WPAR-ekhoz a globális /etc/security/audit/config fájlban.

WPARS szakasz /etc/security/audit/config fájlhoz adásával a globális rendszeradminisztrátor biztosítani tudja a megfigyelendő osztályok listáját a WPAR-hez. Például:

```
WPARS:
<wpar_neve> = <auditclass>, ... <auditclass>
```

Az előző példában a <wpar_neve> a rendszer WPAR neve, és minden auditclass paramétert az osztályok szakaszban kell megadni.

A general, tcpip és lvm osztályokkal rendelkező testwpar WPAR konfiguráláshoz és megfigyeléséhez vegye fel a következő szakaszt a /etc/security/audit/config fájlba:

```
WPARS:
testwpar = general,tcpip,lvm
```

A globális rendszeradminisztrátor el tudja indítani és le tudja állítani a WPAR megfigyelését az **audit** parancs segítségével és a WPAR név megadásával, a következőképp:

```
audit start -@ <wparname1> -@ <wparname2> ...
audit shutdown -@ <wparname1> -@ <wparname2> ...
```

A globális környezetből WPAR objektumok a megfigyelni kívánt objektum abszolút útvonalának megadásával figyelhetők meg. Például a /wpars/wpar1/etc/security/passwd fájl megfigyelési eseményeinek megadásához vegye fel a következő szakaszt a /etc/security/audit/objects fájlba a WPAR partíciót tároló AIX rendszeren:

```
/wpars/wpar1/etc/security/passwd:
     r = "WPAR1_PASSWD_RD"
     w = "WPAR1_PASSWD_WR"
```

A fenti szakaszt a rendszer a megfigyelés indítási (-@ <wpar1>) idejében értelmezi és engedélyezi a wpar1 /etc/security/passwd objektumának megfigyelését. Ezek az attribútumok WPAR1_PASSWD_RD megfigyelési eseményt hoznak létre a /wpars/wpar1/etc/security/passwd fájl minden olvasásakor. Az attribútumok WPAR1_PASSWD_WR megfigyelési eseményt is létrehoznak a fájl minden írásra történő megnyitásakor.

Megjegyzés: Engedélyeznie kell a globális környezet megfigyelését, hogy a globális környezetből engedélyezhesse a WPAR megfigyelést.

Az **auditpr** parancs segítségével előállítható egy megfigyelési jelentés, amely megjeleníti a WPAR nevét. Például:

```
auditpr -v < /audit/trail
```

Megfigyelés az NFS környezetben

Az AIX megfigyelési alrendszer támogatja a beillesztett fájlrendszerek megfigyelését. A beillesztett fájlrendszer konfigurációja a kliensen hasonló a helyi fájlrendszerhez. A megfigyelési műveletek a megfigyelhető beillesztett objektumokon hasonlóak a Megfigyelés áttekintése részben leírt helyi objektumokhoz. A beillesztett fájlrendszerek viselkedését a kliensen és a szerveren a témakör későbbi információi írják le.

Megfigyelés az NFS kliensen

A kliens által a beillesztett fájlrendszerek megfigyelhető objektumain futtatott minden művelet a kliensen kerül naplózásra. Ez érvényes, amennyiben az NFS szerver vagy más NFS kliens nem végez műveleteket az objektumokon, vagy pedig engedélyezni kell a teljes útvonal megfigyelést a kliensen.

További információkért tekintse meg az **audit** parancs főoldalát. Ha a teljes útvonal megfigyelés nem engedélyezett és a fájlt módosítja a szerver vagy más kliensek, akkor a következő megfigyelés eredménye megjósolhatatlan lesz. A viselkedés javítása érdekében indítsa újra a megfigyelést a kliensen. Ha egy fájlrendszer több kliensen is be van illesztve, akkor célszerű a szerveren megfigyelni a műveleteket, hogy megkapja az események pontos naplózását, illetve engedélyezni a teljes útvonal megfigyelést a kliensen.

Megjegyzés: A felülvizsgálati alrendszer konfiguráció nem támogatja a felülvizsgálati napló fájlrendszerének használatát beillesztett NFS fájlrendszerként.

Megfigyelés az NFS szerveren

A beillesztett fájlrendszeren a kliens és a szerver által végrehajtott minden művelet az NFS szerveren kerül naplózásra.

Szerver oldali korlátozások

- Ha az NFS kliens által végrehajtott valamilyen műveletet nem küld el a program a szerverre, akár az NFS gyorsítótárazása, akár az NFS architektúra miatt, akkor a műveletet nem tudja megfigyelni a szerver.

  Például: A fájlrendszer beillesztését követően csak a fájlon végzett első olvasási műveletet figyeli meg a szerver. Az ezt követő olvasási műveletek nem kerülnek naplózásra a szerveren. Ez vonatkozik a fájlok, hivatkozások és könyvtárak olvasási műveleteire is.
- Az ügyfél által végrehajtott műveletek mint **nfsd** kerülnek naplózásra a kiszolgálón, és a **root** felhasználó tartozik hozzájuk felhasználónévként.

Példa

A *File_System* nevű fájlrendszer beillesztésre kerül a kliensen a **mount server:/File_system /mnt** paranccsal. Ha az *A* nevű fájlt a *File_System* fájlrendszeren meg kell figyelni a szerveren, akkor a /File_system/A útvonalat konfigurálni kell a megfigyelési konfigurációs fájlokban.

Ha úgy dönt, hogy a *File_System* fájlrendszeren lévő *A* fájlt a kliensen figyeli meg, akkor az /mnt/A útvonalat be kell állítani megfigyelésre a kliensen.

Ha az *A* fájl a szerveren és a kliensen is be van állítva megfigyelésre, akkor az *A* fájlon a szerver és a kliens által is végrehajtott műveletek a szerveren kerülnek megfigyelésre és naplózásra, a kliens által végrehajtott műveletek pedig a kliensen kerülnek naplózásra.

Az *A* fájlon a kliens által végrehajtott műveletek a művelet vagy parancs neve helyett **nfsd** démonként kerülnek naplózásra.

Egyszerűsített címtárhozzáférési protokoll

Az Egyszerűsített címtárhozzáférési protokoll (LDAP) a címtárak (vagy adatbázisok) információinak helyi vagy távoli hozzáférésének illetve frissítésének szabványos módszerét adja meg egy kliens-szerver modellben.

A protokoll olvasásra, böngészésre és a könyvtárak keresésére van optimalizálva, és eredetileg az X.500 könyvtárhozzáférési protokoll egyszerűsített megjelenítéséhez készült. Az LDAP módszert több fürt használja, ami lehetővé teszi a központi biztonsági hitelesítést és a felhasználói- és csoportinformációkhoz való hozzáférést. A funkcionalitás hálózati környezethez készült, és a hitelesítési-, felhasználói- és csoport információkat tartja összhangban a fürtben.

Az objektumok az LDAP-ban hierarchikus szerkezetben kerülnek tárolásra. Ezt a hierarchikus szerkezetet Címtár információs fának (DIT) nevezzük. A jó könyvtár a DIT szerkezeti tervezésével kezdődik. A DIT-t nagy körültekintéssel kell megtervezni, mielőtt az LDAP-ot hitelesítési módszerként valósítaná meg.

LDAP hitelesítési modul

Az LDAP biztonsági alrendszert az LDAP hitelesítési modul valósítja meg. Alapelvében hasonlít az egyéb betöltési modulokhoz (például NIS, DCE és KRB5). A betöltési modulok az /usr/lib/security/methods.cfg fájlban vannak definiálva.

Az LDAP betöltési modul felhasználói hitelesítést és központosított felhasználó- és csoportkezelő funkcionalitást biztosít az LDAP protokollon keresztül. Az LDAP szerveren definiált felhasználókat be lehet úgy állítani, hogy még akkor is bejelentkezhessenek egy LDAP kliensre, ha az adott kliensen a felhasználók helyi módban nincsenek definiálva.

Az AIX LDAP betöltési modul teljesen integrálva van az AIX operációs rendszerbe. Ha engedélyezte az LDAP hitelesítési modulnak a felhasználói- és csoport információk kiszolgálását, akkor a magasszintű API-k, parancsok és rendszerkezelő eszközök a szokásos módon használhatók. A legtöbb magasszintű parancsnál az **-R** kapcsolóval dolgozhatunk különböző modulokkal. Ha például egy *joe* nevű LDAP felhasználót szeretne létrehozni egy kliensről, akkor írja be a következő parancsot:

```
mkuser -R LDAP joe
```

Megjegyzés: Az LDAP infrastruktúra elvileg korlátlan számú csoporttag létrehozását támogatja. A tesztek során 25.000 csoporttag létrehozására került sor, majd ezután különféle műveletekkel tesztelték a csoportot. Bizonyos archaikus POSIX felületek nem adják vissza a csoportra vonatkozó összes információt. Az ehhez hasonló korlátozásokról tájékozódjon az adott API dokumentációjából.

LDAP alapú hitelesítése:

Az LDAP alapú hitelesítés részét képező egyes entitásokra korlátozások vonatkoznak az AIX rendszeren.

Fontos megjegyzeni, hogy maga az LDAP infrastruktúra nem tesz megszorításokat az adatbázis tartalmára vonatkozóan. Ez a dokumentum a tesztkonfigurációk segítségével megállapított gyakorlati korlátokat ismerteti. Az alábbi korlátok kerültek tesztelésre az LDAP alapú hitelesítéssel AIX operációs rendszeren:

Összes felhasználó maximális száma: A tesztek 500.000 felhasználó létrehozását, és több száz felhasználó egyidejű hitelesítését tartalmazták.

Összes csoport száma maximális száma: A tesztek során egyetlen rendszeren 500 csoport került létrehozásra.

Felhasználók maximális száma csoportonként: A tesztek során 25.000 felhasználót hoztak létre egyetlen csoportban, és különböző műveleteket hajtottak végre ezzel a csoporttal.

Bizonyos archaikus POSIX felületek nem adják vissza a csoportra vonatkozó összes információt. Az ehhez hasonló korlátozásokról tájékozódjon az adott API dokumentációjából. A fenti értékek egy adott teszt eredményeit tükrözik. Nem zárják ki azt, hogy megfelelő erőforrások megléte esetén akár sokkal nagyobb teljesítményű rendszert lehessen építeni.

IBM Tivoli Directory Server biztonsági információs szerver beállítása:

Egy rendszernek LDAP biztonsági információs szerverként való beállításához, amely a hitelesítést, felhasználói- és csoportinformációkat LDAP-on szolgálja ki, először telepítenie kell az LDAP szerver- és kliens csomagokat.

Ha a Védett socket réteg (SSL) szükséges, akkor telepítenie kell a **GSKitV7** csomagot IBM Tivoli Directory Server 6.2 vagy korábbi változathoz, vagy a GSKitV8 csomagot IBM Tivoli Directory Server 6.3 vagy újabb változathoz. A rendszeradminisztrátornak létre kell hoznia egy kulcsot a GSKit kulcskezelési parancsával. Ez a parancs **gsk7ikm** a GSKitV7 csomagban, illetve **ikeyman** a GSKitV8 csomagban. Ha további információkra van szüksége arról, hogy hogyan kell a szervert SSL használatára beállítani, akkor nézze át a Biztonságos kommunikációs SSL-lel részt.

Futtassa az **mksecldap** parancsot a szerver konfigurálásához. Az**mksecldap** parancs kialakítja az LDAP szervert és annak *ldapdb2* háttér adatbázisát, feltölti az LDAP szervert a felhasználói- és csoportinformációkkal a helyi hosztról, valamint beállítja az LDAP szerver adminisztrátor megkülönböztetett nevét (DN) és jelszavát. Nem kötelezően beállítja az SSL-t a kliens-szerver kommunikációhoz. Továbbá az **mksecldap** parancs hozzáad egy bejegyzést az /etc/inittab fájlhoz, hogy az LDAP szerver minden rendszerindításkor elinduljon.

Az AIX felhasználók és csoportok az LDAP szerveren az alábbi sémák valamelyikével van tárolva:

AIX séma
: Az aixAccount és aixAccessGroup objektumosztályokat tartalmazza. Ez a séma az AIX felhasználók és csoportok teljes attribútumkészletét tartalmazza.

RFC 2307 séma
: A posixAccount, shadowAccount és posixGroup objektumosztályt tartalmazza; számos szállító címtár-terméke használja. Az RFC 2307 séma csak az AIX által használt attribútumok egy kis részét definiálja.

RFC2307AIX séma
: A posixAccount, a shadowAccount és posixGroup, valamint az aixAuxAccount és aixAuxGroup objektumosztályokat tartalmazza. Az aixAuxAccount és aixAuxGroup objektumosztály azokat az attribútumokat biztosítja, amelyeket az AIX használ, de amelyek nincsenek definiálva az RFC 2307 sémában.

Az RFC2307AIX sématípus használata felhasználók és csoportok esetén nagyon ajánlott. Az RFC2037AIX sématípus az RFC 2307 szabványnak teljesen megfelelő, további attribútumokkal, hogy még több AIX felhasználókezelési funkcionalitást támogasson. Az IBM Tivoli Directory Server szerver RFC2307AIX séma konfigurációval nem csupán AIX LDAP klienseket, hanem más, RFC 2307 szabványnak megfelelő UNIX és Linux LDAP klienseket is támogat.

A felhasználói- és csoportinformációk teljes köre egy közösAIX fa alatt van tárolva (utótag). Az alapértelmezett utótag a "cn=aixdata". Az **mksecldap** parancs a **-d** kapcsolóval fogadja a felhasználó által megadott utótagot. A felhasználó, csoport, azonosító stb. számára létrehozásra kerülő al-fákat a sectoldif.cfg konfigurációs fájl határozza meg. További információkat a sectoldif.cfg fájlban talál.

Az AIX fa ACL (hozzáférés-felügyeleti lista) által védett. Az alapértelmezett ACL csak annak a felhasználónak ad adminisztrátori jogosultságot, aki adminisztrátorként került megadásra az **-a** parancs kapcsolóval. További jogosultságokat lehet adni egy proxy azonosságnak az **-x** és **-X** parancs kapcsoló használatával. A kapcsolók létrehozzák a proxy azonosságot és az /etc/security/ldap/proxy.ldif.template fájl alapján beállítják az azonosság hozzáférési jogosultságait. A proxy azonosság létrehozása lehetővé teszi az LDAP kliensek számára a szerver hozzárendelését az adminisztrátori azonosság használata nélkül, ami korlátozza a kliens adminisztrátor felhatalmazásokat az LDAP szerveren.

Az **mksecldap** parancsot futtathatja olyan LDAP szerveren, amely más célokra, például felhasználói azonosító kikereséséhez lett beállítva. Ebben a példában az **mksecldap** hozzáadja az AIX fát feltölti azt az AIX biztonsági információkkal a meglévő LDAP szerverre. Ez a fa ACL által védett, más meglévő fáktól függetlenül.

Megjegyzés: Végezzen biztonsági mentést a meglévő LDAP szerverről, mielőtt futtatná az **mksecldap** parancsot és kibontaná a szervert egy AIX biztonsági információs szerverre.

Az LDAP biztonsági információs szerver sikeres beállítása után ugyanezt a hosztot beállíthatja kliensként az LDAP felhasználók és csoportok kezeléséhez, és engedélyezheti LDAP felhasználók számára a bejelentkezést erre a szerverre.

Ha az LDAP biztonsági információs szerver beállítása nem sikerül, akkor az **mksecldap** parancs **-U** kapcsolóval futtatásával visszavonhatja a beállítást. A parancs visszaállítja az ibmslapd.conf (vagy slapd.conf vagy slapd32.conf) fájlt a beállítás előtti állapotra. A sikertelen beállítási kísérlet után még az **mksecldap** parancs ismételt futtatása előtt futtassa az **mksecldap** parancsot az **-U** kapcsolóval. Ellenkező esetben beállítási információk maradhatnak a konfigurációs fájlokban, amelyek további sikertelenségeket okozhatnak. A biztonság kedvéért a visszavonás kapcsoló nem csinál semmit az adatbázissal és az adatbázis adataival, mivel az adatbázis már az **mksecldap** parancs futtatása előtt is létezhetett. Ha az adatbázist az **mksecldap** parancs hozta létre, akkor távolítsa el manuálisan. Ha az **mksecldap** parancs egy meglévő adatbázishoz adott hozzá adatokat, akkor döntse el, hogy milyen lépésekkel állítja helyre a sikertelen beállítást.

Kapcsolódó fogalmak:

Biztonságos kommunikáció SSL-lel
Az LDAP kliens és a szerver között használt hitelesítés típusától függően a jelszavak titkosított módon (unix_auth) vagy sima szövegként (ldap_auth) kerülnek elküldésre. A biztonsági kockázatok kivédésére használjon Védett socket réteget (SSL) még a titkosított jelszavak hálózaton - vagy egyes esetekben Interneten - keresztüli küldéséhez is. Az AIX tartalmaz olyan SSL csomagokat, amelyek biztonságos kommunikációs biztosítanak a címtár szerverek és a kliensek között.

Kapcsolódó tájékoztatás:

mksecldap parancs

LDAP kliens beállítása:

Ha a klienseket úgy szeretné beállítani, hogy a hitelesítéshez és a felhasználói/csoport információkhoz az LDAP-ot használják, akkor győződjön meg róla, hogy az LDAP kliens csomag minden kliensre telepítve van. Ha Védett socket réteg (SSL) szükséges, akkor a GSKit csomagot telepíteni kell, létre kell hozni egy kulcsot, és hozzá kell adni ehhez a kulcshoz a LDAP szerver SSL kulcs igazolását.

Az LDAP szerver beállításához hasonlóan a kliens beállítását is el lehet végezni az **mksecldap** paranccsal. Ha azt szeretné, hogy a kliens kapcsolatba lépjen az LDAP biztonsági információs szerverrel, akkor a szerver nevét meg kell adni a beállítás során. A szerver kötési megkülönböztetett nevére és jelszavára szintén szüksége van a kliens hozzáférésnek a szerver AIX fájához. Az **mksecldap** parancs a szerver kötési megkülönböztetett nevét, jelszavát, szervernevét és AIX fa megkülönböztetett nevét elmenti a szerverre, az SSL kulcs elérési útját és jelszavát valamint az egyéb konfigurációs attribútumokat pedig az /etc/security/ldap/ldap.cfg fájlba.

Az **mksecldap** parancs elmenti a kötési jelszót és az SSL kulcs jelszót (SSL beállítása esetén) az /etc/security/ldap/ldap.cfg fájlba titkosított formátumban. A titkosított jelszavak rendszerspecifikusak, és a **secldapclntd** démon csak azon a rendszeren használhatja, amelyen előállításra kerültek. A **secldapcIntd** démon csak a /etc/security/ldap/ldap.cfg fájlból származó nyílt szövegű vagy titkosított jelszót használhatja.

A kliens beállítása közben az **mksecldap** parancsnak több szervert is meg lehet adni. Ebben az esetben a kliens a megadott sorrendben lépesít kapcsolatot a szerverekkel, és az első sikeresen csatlakoztatott szerverrel létesít kötést. Ha a kliens és a szerver kapcsolatában hiba jelentkezik, akkor a kliens ugyanígy próbál meg ismét kapcsolatot létesíteni. A LDAP biztonsági megvalósítása nem támogatja a hivatkozást. Fontos a többszörözött szervereket szinkronban tartani.

A kliens egy kliens démonon keresztül (**secldapcIntd**) kommunikál az LDAP biztonsági információs szerverrel. Ha az LDAP modul engedélyezve van a kliensen, akkor a magasszintű parancsok a démonhoz kerülnek átirányításra a könyvtár API-kon keresztül az LDAP-ban definiált felhasználóknál. A démon egy ideiglenes tárolóban tárolja a kért LDAP bejegyzéseket. Ha egy kérést nem lehet kiszolgálni az ideiglenes tárolóból, akkor a démon lekérdezi a szervert, frissíti az ideiglenes tárolót, és visszaadja az információkat a hívónak.

Egyéb finomhangolási beállítások is megadhatók az **mksecldap** parancsnak a kliens beállítása közben - a démon által használt szálak száma, az ideiglenes tároló bejegyzés mérete, az ideiglenes tároló lejárati időkorlátja. Ezek a beállítások a tapasztalt felhasználók számára készültek. A legtöbb környezetben az alapértelmezett értékek elegendőek.

A kliens beállításának utolsó lépésében az **mksecldap** parancs elindítja a kliensoldali démont, és egy bejegyzést ad hozzá az /etc/inittab fájlhoz, hogy a démon minden újraindításkor elinduljon. A beállítás sikerességéről a **secldapcIntd** démonfolyamat ellenőrzésével győződhet meg az **ls-secldapclntd** parancs segítségével. Ha az LDAP biztonsági információs szerver be van állítva és fut, akkor a sikeres beállítás után ez a démon futni fog.

A szervert a kliens előtt kell beállítani. A kliens beállítása a szerveren lévő átállított adatoktól függ. A kliens beállításához tegye a következőket:

1. Telepítse az IBM Tivoli Directory Server kliens fájlkészletet az AIX operációs rendszeren.
   - IBM Tivoli Directory Server 5.2 esetén telepítse az ldap.client fájlkészletet.
   - IBM Tivoli Directory Server 7.1 esetén telepítse az idsldap fájlkészletet.
2. Az LDAP kliens beállításához futtassa az alábbi parancsot:

   ```
   # mksecldap -c -h server1.ibm.com -a cn=admindn -p adminpwd -d cn=basedn
   ```

   Cserélje le a fenti értékeket a környezetnek megfelelően.

Kapcsolódó tájékoztatás:

mksecldap parancs

secldapclntd parancs

Kliensfelkészítés LDAP hálózati csoportokhoz (netgroup):

A hálózati csoportok használhatók a NIS-LDAP (a névfeloldási metódus) részeként.

Az LDAP hálózati csoportok kliensre felkészítéséhez végezze el az alábbi lépéseket:

1. Telepítsen és állítson be LDAP alapú felhasználói csoportkezelést az alábbi részben leírt módon: “LDAP kliens beállítása” oldalszám: 152.

Ha a hálózati csoport beállítás nem befejeződött be, akkor a rendszer minden LDAP által megadott felhasználót megjelenít. Ha például az *nguser* az LDAP szerveren megadott *mygroup* hálózati csoporthoz tartozó felhasználó, akkor az lsuser -R LDAP nguser kiírja a felhasználót.

2. A hálózati csoport funkció engedélyezéséhez a /usr/lib/security/methods.cfg fájlban lévő LDAP moduldefiníciónak tartalmaznia kell egy netgroup értékű options attribútumot. Szerkessze a /usr/lib/security/methods.cfg fájlt és az LDAP szakaszhoz adja hozzá az options = netgroup sort. Ez az LDAP betölthető modult hálózati csoport képes betöltési modulként jelöli. Például:

   ```
   LDAP:
         program = /usr/lib/security/LDAP
         program_64 =/usr/lib/security/LDAP64
         options = netgroup
   ```

   Ezután az lsuser -R LDAP nguser, lsuser nguser vagy lsuser -R LDAP -a ALL parancs nem jelenít meg egy felhasználót sem. Az LDAP mostantól erről a kliensről csak a hálózati csoportok számára számít elérhetőnek, és még a kliens egyetlen hálózati csoportja számára sincs engedélyezve a hozzáférés.

3. Szerkessze a /etc/passwd fájlt és fűzzön hozzá egy hálózati csoporthoz tatozó sort, amelynek hozzá kell férnie a rendszerhez. Ha például a *mygroup* a hálózati csoport a kívánt felhasználót tartalmazó LDAP szerveren, akkor fűzze hozzá az alábbit:

   ```
   +@mygroup
   ```

4. Szerkessze a /etc/group fájlt és fűzzön hozzá egy +: sort a csoportok NIS kikeresésének engedélyezéséhez:

   ```
   +:
   ```

   Az lsuser nguser parancs futtatása a visszaadja a felhasználót, mivel az *nguser* a *mygroup* hálózati csoportban van.

   Az lsuser -R LDAP nguser parancs nem találja meg a felhasználót, de az lsuser -R compat nguser parancs igen, mivel a felhasználó **compat** felhasználó.

5. Ahhoz, hogy a hálózati csoport felhasználók hitelesíteni tudják magukat a rendszerhez, az AIX hitelesítési mechanizmusnak ismernie kell a használandó módszert. Ha a /etc/security/user fájl alapértelmezett szakasza tartalmazza a SYSTEM = compat értéket, akkor a /etc/passwd fájlhoz hozzáadott netgroup felhasználói hitelesíteni tudják magukat. Másik lehetőség a felhasználók egyenkénti beállítása, amelynek során a kívánt felhasználók szakaszait kézzel kell hozzáadni a /etc/security/user fájlhoz. Az *nguser* példa szakasza:

   ```
   nguser:
           SYSTEM = compat
           registry = compat
   ```

   Az engedélyezett hálózati csoportok felhasználói hitelesíthetik magukat a rendszerhez.

   A hálózati csoport szolgáltatás engedélyezése a következő feltételeket is aktiválja:

   - A /etc/security/user fájlban az LDAP nyilvántartás tagjaként megadott felhasználók (registry=LDAP és SYSTEM="LDAP" beállítással rendelkeznek) nem tudják hitelesíteni magukat LDAP felhasználókként. Ezek a felhasználók **nis_ldap** felhasználók és natív NIS hálózati csoport tagságot igényelnek.
   - A nyilvántartás compat jelentése kibővítésre került, hogy tartalmazza a hálózati csoportot használó modulokat. Ha például az LDAP modul hálózati csoport támogatással rendelkezik, akkor a compat tartalmazza a fájlokat valamint a NIS és LDAP nyilvántartásokat. Ezen modulokból lekért felhasználók compat nyilvántartásértékkel rendelkeznek.

Kapcsolódó információk

- NFS fájlrendszer exports fájlja dokumentum
- TCP/IP .rhosts fájlformátuma dokumentum
- TCP/IP hosts.equiv fájlformátuma dokumentum

Támogatott LDAP szerverek:

Az AIX LDAP alapú felhasználó- és csoportkezelés támogatja az IBM Tivoli Directory Servereket, az RFC 2307-nek megfelelő nem IBM szervereket és a Microsoft Active Directory szervereket.

IBM Tivoli Directory Server

Erősen ajánlott, hogy az AIX felhasználó/csoport kezelésének beállítását az IBM Tivoli Directory Server használatával valósítsa meg. Az IBM Tivoli Directory Server felhasználó- és csoportkezelési célú beállításával kapcsolatban további információkért lésd: IBM Tivoli Directory Server biztonsági információs szerver beállítása.

Nem IBM Directory Serverek

Az AIX különböző címtárszervereket támogat, amelyek felhasználói és csoportjai az RFC 2307 sémával vannak megadva. Amikor ilyen szerverekhez LDAP kliensként van konfigurálva, akkor az AIX a szervereket ugyanolyan módon használja, mint egy IBM Tivoli Directory Server szervert RFC 2037 sémával. Ezeknek a szervereknek támogatniuk kell az LDAP 3-as változatú protokollt.

Mivel az RFC 2307 séma az AIX által használható felhasználó- és csoportattribútumok egy részhalmazát adja csak meg, néhány AIX felhasználó- és csoportkezelési funkció nem hajtható végre, ha az AIX ilyen LDAP szerver használatára van beállítva (például felhasználói jelszó visszaállítás kikényszerítése, jelszóelőzmény, felhasználónkénti erőforráskorlát, bejelentkezés-vezérlés bizonyos rendszerekhez AIX hostsallowedlogin és hostsdeniedlogin attribútumokon keresztül, képesség és így tovább).

Az AIX nem támogatja a RFC 2307-nek nem megfelelő címtárszervereket. Azonban beállítható az AIX az RFC 2307-nek nem megfelelő szerverek kezelésére, de csak olyanokéra, amelyek felhasználói és csoportjai az összes szükséges UNIX attribútummal vannak megadva. Az AIX által igényelt felhasználó- és csoportattribútumok minimális halmaza az RFC 2307-ben megadott halmaz. Az ilyen címtárszerverek támogatása kézi konfigurációt igényel. Az AIX sémaleképezési mechanizmust biztos ebből a célból. A sémafájlformátummal és a sémafájlhasználattal kapcsolatos további információkat az LDAP attribútumleképezési fájlformátum rész tartalmaz.

Microsoft Active Directory

Az AIX támogatja a Microsoft Active Directory (AD) használatát LDAP szerverként felhasználó- és csoportkezeléshez. Az AD szervernek rendelkeznie kell telepített UNIX támogatási sémával. Az AD UNIX támogatási sémája a Microsoft Service For UNIX (SFU) csomagból származik. Minden SFU változat kicsit különböző felhasználó- és csoportséma-meghatározásokkal rendelkezik, mint a megelőző változat. Az AIX támogatja az Windows 2000 és 2003 rendszeren futó AD-t SFU séma 3.0 és 3.5 változattal, és a Windows 2003 R2 rendszeren futó AD-t a beépített UNIX sémával.

A UNIX és Windows rendszerek felhasználó- és csoportkezelési elétérései miatt nem minden AIX parancs működik az LDAP felhasználókra vonatkozóan, ha a szerver AD. Az **mkuser** és **mkgroup** parancs szintén nem működik. A legtöbb felhasználó- és csoportkezelési parancs működik, az azonosságnak adott hozzáférési jogoktól függően, amellyel az AIX az AD-hez kötést végzi. Ezek a parancsok: **lsuser**, **chuser**, **rmuser**, **lsgroup**, **chgroup**, **rmgroup**, **id**, **groups**, **passwd** és **chpasswd**.

Az AIX két felhasználóhitelesítési mechanizmust támogat Windows szervereken: LDAP hitelesítést és Kerberos hitelesítést. Az AIX mindkét mechanizmus esetén támogatja az LDAP címtáron keresztüli felhasználóazonosítást az AD-vel szemben, anélkül, hogy az AIX rendszeren megfelelő felhasználói fiókra lenne szükség.

Az AIX operációs rendszer beállítása Active Directory együttműködéshez LDAP címtáron keresztül:

Az AIX támogatja a Microsoft Active Directory (AD) használatát LDAP szerverként felhasználó- és csoportkezeléshez. Ehhez az AD szervernek telepített UNIX támogatási sémával kell rendelkeznie.

Az adminisztrátorok az**mksecldap** paranccsalkonfigurálhatják az AIX rendszert az AD szerveren, ugyanúgy, ahogyan egy IBM Tivoli Directory Server esetében. Az **mksecldap** parancs elrejti a konfiguráció részleteit a folyamat egyszerűsítése érdekében. Az **mksecldap** parancs futtatása előtt az AIX AD szerveren való beállítása:

1. Az AD szervernek rendelkeznie kell telepített UNIX támogatási sémával.
2. Az AD szervernek tartalmaznia kell UNIX rendszert használni képes felhasználókat.

A UNIX AD szerveren telepítésével és az AD felhasználók UNIX támogatással való engedélyezéséhez tekintse meg a kapcsolódó Microsoft dokumentációt.

Az AD séma gyakran rendelkezik több attribútummeghatározással ugyanahhoz a UNIX attribútumhoz (például több felhasználói jelszó és csoporttag-meghatározás). Annak ellenére, hogy az AIX ezek többségét támogatja, az átgondolást és tervezést körültekintően kell elvégezni a használandó meghatározások kiválasztásakor. Az ütközés elkerülése érdekében ajánlatos, hogy ugyanazon AD-t megosztó AIX és más nem AIX rendszerek ugyanazt a meghatározást használják.

Active Directory jelszóattribútum kiválasztása:

Az AIX két hitelesítési mechanizmust támogat, ezek a következők: **unix_auth** és l**dap_auth**.

unix_auth esetén a Microsoft Active Directory-ban (AD) lévő jelszót titkosítani kell. A hitelesítés során a titkosított jelszó lekérésre kerül az AD-ből és összehasonlításra kerül a felhasználó által megadott jelszó titkosított formátumával. A hitelesítés sikeres, ha a két jelszó megegyezik. **ldap_auth** módban az AIX egy LDAP kötési művelettel hitelesíti a felhasználót a szerver felé a felhasználó azonosságával és a megadott jelszóval. A felhasználó hitelesítésre kerül, ha a kötési művelet sikeres. Az AD több felhasználói jelszó attribútumot támogat. Eltérő AIX hitelesítési mód szükséges az eltérő AD felhasználói jelszó attribútumhoz.

unix_auth mód

A következő AD jelszóattribútumok használhatók a **unix_auth** módhoz:

- **userPassword**
- **unixUserPassword**
- **msSFU30Password**

A jelszókezelés AIX rendszeren bonyolult lehet, mivel az AD több jelszóattribútummal rendelkezik. Annak meghatározása, hogy a UNIX klienseknek mely jelszókezelési attribútumokat kell használniuk, nem egyértelmű. Az AIX LDAP attribútum leképezési képesség lehetővé teszi a jelszókezelés személyre szabását az igényeknek megfelelően.

Alapértelmezésben az AIX az **msSFU30Password** attribútumot használja Windows 2000 és 2003 rendszeren futó AD-hez, illetve a **userPassword** attribútumot Windows 2003 R2 rendszeren. Ha másik jelszó kerül felhasználásra, akkor módosítani kell az /etc/security/ldap/sfu30user.map fájlt (vagy az /etc/security/ldap/sfur2user.map fájlt, ha az AD Windows 2003 R2 rendszeren fut). Keresse meg az **spassword** szóval kezdődő sort és módosítsa a sor harmadik mezejét a kívánt AD jelszóattribútum-névre. További információkat az LDAP attribútumleképezési fájlformátum rész tartalmaz. Futtassa az **mksecldap** parancsot az AIX LDAP kliens módosítás utáni beállításához. Ha az AIX LDAP kliens már be van állítva, akkor futtassa a **restart-secldapclntd** parancsot a **secldapclntd** démon újraindításához a módosítás átvétele érdekében.

unix_auth módban elképzelhető, hogy a jelszó nincs szinkronban a Windows és UNIX rendszer között, ezáltal az egyes rendszerekhez eltérő jelszó tartozik. Ez akkor történik, ha a jelszót AIX és Windows rendszer között módosítja, mivel a Windows az **uncodepwd** jelszóattribútumot használja. Az AIX **passwd** parancs vissza tudja állítani a UNIX jelszót, hogy ugyanaz legyen, mint a Windows jelszó, de az AIX nem támogatja automatikusan a Window jelszó módosítását a UNIX jelszó AIX rendszerről történő módosítása esetén.

ldap_auth mód

Az Active Directory szintén rendelkezik **unicodepwd** jelszóattribútummal. A jelszóattribútumot a Windows rendszer használja a Windows felhasználók hitelesítéséhez. AD-hez kötési műveletben a **unicodePwd** jelszót kell használni. A **unix_auth** mód alatt említett jelszavak egyike sem működik kötési művelet esetén. Ha az **ldap_auth** beállítás a parancssorból lett megadva, akkor az **mksecldap** parancs leképezi a jelszóattribútumot az AD **unicodePwd** attribútumára a klienskonfigurációnál kézi lépés nélkül.

Az AIX jelszavak **unicodePwd** attribútummal való leképezésével az AD-ben megadott felhasználók Windows és AIX rendszerre ugyanazzal a jelszóval léphetnek be. AIX vagy Windows rendszerről történő jelszó-visszaállítás AIX és Windows rendszer esetén is érvényes.

Active Directory csoporttag-attribútum kiválasztása:

A Microsoft Service for UNIX megadja a **memberUid**, **msSFU30MemberUid**, és **msSFU30PosixMember** csoporttag-attribútumokat.

A **memberUid** és a **msSFU30MemeberUid** attribútum elfogadja a felhasználói fiók neveket, az **msSFU30PosixMember** azonban csak a teljes megkülönböztetett neveket fogadja el. Az AD-ben megadott *foo* felhasználói fiók esetén (*bar* vezetéknévvel) például:

- **memberUid: foo**
- **msSFU30MemberUid: foo**
- **msSFU30PosixMember: CN=foo bar,CN=Users,DC=austin,DC=ibm,DC=com**

Az AIX operációs rendszer ezen attribútumok mindegyikét támogatja. A használandó attribútum meghatározásához lépjen kapcsolatba az AD adminisztrátorral. Alapértelmezésben az **mksecldap** parancs beállítja az AIX rendszert, hogy az **msSFU30PosixMember** attribútumot Windows 2000 és 2003 rendszeren, az **uidMember** attribútumot pedig Windows 2003 R2 rendszeren futó AD-n használja. Az ilyen kiválasztást az AD viselkedés okozza, mivel az AD ezt az attribútumot akkor választja ki, amikor egy felhasználót csoporthoz ad Windows rendszeren. Az üzleti stratégia igényelheti nem alapértelmezett csoporttag-attribútum használatát több platform támogatása érdekében.

Ha különböző csoporttag-attribútum szükséges, akkor módosíthatja a leképezést a csoportleképezési fájl szerkesztésével. Az AD csoportleképezési fájlja a Windows 2000 és 2003 rendszeren futó /etc/security/ldap/sfu30group.map, illetve az /etc/security/ldap/sfur2group.map Windows 2003 R2 rendszeren. Keresse meg a **users** szóval kezdődő sort és cserélje le a harmadik mezőt a csoporttaghoz a kívánt attribútumnévre. További információkat az LDAP attribútumleképezési fájlformátum rész tartalmaz. Futtassa az **mksecldap** parancsot azAIX LDAP kliens konfigurálásához a módosítás után, vagy ha az AIX kliens már konfigurálva van, akkor futtassa a**restart-secldapclntd** parancsot a **secldapclntd** démon újraindításához,hogy az magába vegye a módosítást.

Több szervezeti egység:

Az AD szerver több megadott szervezeti egységgel rendelkezhet, amelyek mindegyike felhasználók halmazát tartalmazza.

A legtöbb Windows AD felhasználó a **cn=users,...** részfában van megadva, de néhány felhasználó máshol is megadható. Az AIX több alap DN szolgáltatás használható egy ilyen AD szerverhez. További információkat a Több alap DN támogatása rész tartalmaz.

Kerberos hitelesítés Windows szerverekhez:

Az LDAP hitelesítési mechanizmuson felül az AIX operációs rendszer támogatja a felhasználóhitelesítést Kerberos protokollal Windows szerverekhez is.

Az AIX operációs rendszer támogatja a Kerberos hitelesítést Windows KDC és LDAP azonosításhozWindows Active Directory esetén a KRB5ALDAP összetett betöltési modul létrehozásával. Mivel a felhasználóazonosítási információk a Microsoft Active Directory címtárból kerülnek lekérésre, nem kell létrehoznia a megfelelő felhasználói fiókokat az AIX operációs rendszeren.

LDAP felhasználókezelés:

Az LDAP biztonsági információs szerveren található felhasználókat és csoportokat bármely LDAP kliensről kezelheti magasszintű parancsokkal.

A legtöbb magasszintű parancs rendelkezik az **-R** kapcsolóval, amellyel kezelheti az LDAP-ot használó felhasználókat és csoportokat, valamint az egyéb hitelesítési modulokat - DCE, NIS és KRB5. Az **-R** kapcsoló használatáról további információkat az egyes felhasználó- vagy csoportkezelő parancsoknál talál.

Ha engedélyezni szeretné egy felhasználónak az LDAP-on keresztüli hitelesítést, akkor a **chuser** paranccsal módosítsa a felhasználó **SYSTEM** attribútumának értékét LDAP-ra. A **SYSTEM** attribútum értékének megadott szintaxissal való beállításával a felhasználók több betöltési modulon keresztül is hitelesíthetők (pl.: compat és LDAP). A felhasználók hitelesítésének beállításáról további információkat a “Felhasználó hitelesítése” oldalszám: 69 részben és az /etc/security/user fájlban megadott **SYSTEM** attribútum szintaxisában talál.

Egy felhasználó úgy válhat LDAP felhasználóvá, hogy a kliens indításakor az **mksecldap** parancsot a **-u** kapcsolóval futtatja az alábbi formátumok valamelyikében:

1. Futtassa a parancsot:

   ```
   mksecldap -c -u
   felhasználó1,felhasználó2,...
   ```

   ahol a *felhasználó1,felhasználó2,...* a felhasználók listája. A listában szereplő felhasználók lehetnek helyileg vagy távolról LDAP segítségével definiált felhasználók. A parancs a fenti felhasználók szakaszaiban a **SYSTEM** attribútumot LDAP értékre állítja az /etc/security/user fájlban. Az ilyen felhasználókat csak LDAP-on keresztül lehet hitelesíteni. A listában szereplő felhasználóknak léteznie kell az LDAP biztonsági információs szerveren. Ellenkező esetben nem tudnak bejelentkezni erről a hosztról. Az **chuser** paranccsal módosítsa a **SYSTEM** attribútumot, és engedélyezze a többféle módszeren keresztüli hitelesítést (pl: helyi és LDAP).
2. Futtassa a

   ```
   mksecldap -c -u ALL
   ```

   parancsot.Ez a parancs az összes helyileg definiált felhasználó felhasználói szakaszában LDAP-ra állítja a **SYSTEM** attribútumot az /etc/security/user fájlban. Az ilyen felhasználókat csak LDAP-on keresztül lehet hitelesíteni. A helyileg definiált felhasználóknak léteznie kell az LDAP biztonsági információs szerveren. Ellenkező esetben nem tudnak bejelentkezni erről a hosztról. Azok a felhasználók nem tudnak bejelentkezni erről a hosztról, amelyek az LDAP szerveren definiálva vannak ugyan, de helyileg nem. Ha az LDAP által távolról definiált felhasználó számára engedélyezni szeretné a bejelentkezést a hosztról, akkor a **chuser** paranccsal állítsa a felhasználó **SYSTEM** attribútumát LDAP értékre.

Vagy engedélyezze minden LDAP felhasználónak - függetlenül attól, hogy definiálva van-e helyileg vagy sem - az LDAP-on keresztüli hitelesítést a helyi hoszton. Módosítsa az /etc/security/user fájl alapértelmezett szakaszát LDAP használatára. Minden olyan felhasználónak az alapértelmezett szakaszban definiált attribútumot kell követnie, akinek nincs saját **SYSTEM** attribútuma definiálva. Ha például az alapértelmezett szakasz a "SYSTEM = "compat"" beállítást tartalmazza, akkor ennek "SYSTEM = "compat OR LDAP"" beállításra módosítása lehetővé tesz ezen felhasználók AIX rendszeren vagy LDAP protokollon keresztüli hitelesítését. Az alapértelmezett szakasz "SYSTEM = "LDAP"" beállításra módosításával ezek a felhasználók csak LDAP protokollon keresztül kerülnek hitelesítésre. Azokat a felhasználókat nem érinti az alapértelmezett szakasz módosítása, akiknek saját beállított **SYSTEM** attribútumuk van.

Több alap DN támogatása:

Az AIX támogatja a több alap megkülönböztetett nevet (DN). Legfeljebb 10 alap DN adható meg minden entitáshoz az /etc/security/ldap/ldap.cfg fájlban.

Az alap megkülönböztetett nevek prioritással rendelkeznek az /etc/security/ldap/ldap.cfg fájlbeli sorrendjükben. Az AIX parancsok általi művelet több alap DN esetén az alap DN prioritás szerint kerül feldolgozásra a következő viselkedéssel:

- Az alap DN lekérdezési művelete (például az **lsuser** parancs által) a prioritás szerint kerül végrehajtása, amíg a rendszer egyező fiókot nem talál, vagy hibát ad vissza, ha egyik alap megkülönböztetett név sem megfelelő. Az összes lekérdezésekor az összes alap DN összes fiókja visszadásra kerül.
- A módosítás művelet (például a **chuser** paranccsal) az első egyező fiókig kerül végrehajtásra.
- A törlés művelet (például az **rmuser** paranccsal) az első egyező fiókig kerül végrehajtásra.
- A létrehozás művelet (például az **mkuser** paranccsal) csak az első alap megkülönböztetett néven kerül végrehajtásra. Az AIX nem támogatja fiókok létrehozását más alap megkülönböztetett nevekhez.

A címtárszerver adminisztrátorának felelőssége ütközésmentes fiókadatbázis fenntartása. Ha ugyanahhoz a fiókhoz több meghatározás létezik, mindegyik külön részfa alatt, akkor csak az első fiók látható az AIX számára. A keresés művelet csak az első megfelelő fiókot adja vissza. Ehhez hasonlóan a módosítás vagy törlés művelet is csak az első megfelelő fiókon kerül végrehajtásra.

Az **mksecldap** parancs LDAP kliens beállítása esetén megkeresi az alap megkülönböztetett nevet minden entitáshoz és elmenti az /etc/security/ldap/ldap.cfg fájlba. Ha több alap DN áll rendelkezésre az LDAP szerveren egy entitáshoz, akkor az **mksecldap** parancs véletlenszerűen használja azok egyikét. Ahhoz, hogy az AIX több alap megkülönböztetett nevet kezeljen, szerkeszteni kell az /etc/security/ldap/ldap.cfg fájlt az **mksecldap** parancs sikeres végrehajtása után. Keresse meg a megfelelő alap DN meghatározást és szükség esetén adjon hozzá további alap megkülönböztetett neveket. Az AIX legfeljebb 10 alap megkülönböztetett nevet támogat minden entitáshoz. A továbbiak figyelmen kívül maradnak.

Az AIX a felhasználó által megadott szűrőt, valamint a keresési hatókört is támogatja minden alap megkülönböztetett névhez. Az alap DN rendelkezhet saját szűrővel és hatókörrel, amely különbözhet a partner alap megkülönböztetett névtől. A szűrők segítségével megadható az AIX számára látható fiókok halmaza.

Csak a szűrőt kielégítő fiókok láthatók az AIX számára.

SSL beállítása az LDAP szerveren:

LDAP szerveren a Védett socket réteg (SSL) beállításához telepítse az LDAP titkosítási fájlkészleteket és a **GSKit** fájlkészleteket, amivel lehetővé teszi a szerver titkosítás támogatást. Ezek a fájlkészletek megtalálhatók az AIX bővítőcsomagban.

Az alábbi lépések végrehajtásával engedélyezze az SSL támogatást az IBM Directory szerver hitelesítéshez.

1. Telepítse a IBM Tivoli Directory Server **GSKit** for IBM Tivoli Directory Server v6.2 vagy a **GSKitv8** for IBM Tivoli Directory Server v6.3 csomagot, ha még nincs telepítve.
2. Állítsa elő az IBM címtár szerver privát kulcsát és szerver igazolását a megfelelő GSKit kulcskezelési segédprogrammal. Az IBM Tivoli Directory Server 6.2 változata esetén használja a **gsk7ikm** segédprogramot, IBM Tivoli Directory Server 6.3 vagy újabb változat esetén használja az ikeyman eszközt. A szerver igazolását egy kereskedelmi hitelesítő szervezet (CA), például a VeriSign írta alá, vagy lehet, hogy saját aláírású a GSKit kulcskezelési által. A CA nyilvános igazolását (vagy a saját aláírású igazolást) terjeszteni is kell a kliens alkalmazás kulcsadatbázis fájlja felé.
3. Tárolja el a szerver kulcsadatbázis fájlját és a kapcsolatos jelszó gyűjtőfájlt a szerveren. A kulcs adatbázis alapértelmezett elérési útja tipikusan az /usr/ldap/etc könyvtár.
4. A szerver beállításához futtassa a következő parancsot, ahol **mykey.kdb** a kulcsadatbázis, *keypwd* a jelszó a kulcsadatbázishoz:

```
# mksecldap -s -a cn=admin -p pwd -S rfc2307aix -k /usr/ldap/etc/mykey.kdb -w keypwd
```

SSL beállítása az LDAP kliensen:

LDAP kliensen SSL használatához telepítse az ldap.max_crypto_client és GSKit fájlkészleteket azAIX bővítőcsomagból.

Az alábbi lépések végrehajtásával engedélyezze az LDAP SSL támogatását, miután a szerveren már engedélyezte az SSL-t.

1. A gsk7ikm futtatásával hozza létre a kulcs adatbázist minden egyes kliensen.
2. Másolja át a szerver igazolást minden egyes kliensre. Ha a szerver SSL saját aláírású igazolást használ, akkor az igazolást először ki kell exportálni.
3. A kliens rendszereken a gsk7ikm futtatásával importálja be a szerver igazolást a kulcs adatbázisba.
4. Engedélyezze az SSL-t minden kliensen:
   ```
   # mksecldap -c -h servername -a adminDN -p pwd -k /usr/ldap/etc/mykey.kdb -p keypwd
   ```

   Ahol a /usr/ldap/etc/mykey.kdb a kulcsadatbázis teljes elérési útja, a *keypwd* pedig a kulcs jelszava. Ha a kulcs jelszót nem írja be a parancssorból, akkor a rendszer az adott könyvtár titkosított jelszó fájlját használja. A titkosított fájl nevének meg kell egyeznie a kulcs adatbázis nevével, de a kiterjesztésének **.sth**-nak kell lennie (például sajátkulcs.sth).

LDAP hoszt hozzáférés felügyelet:

Az AIX felhasználói szintű hoszt hozzáférési (bejelentkezési) felügyeletet biztosít a rendszer számára. Az adminisztrátor a felhasználók **SYSTEM** attribútumának LDAP értékre állításával beállíthatja úgy az LDAP felhasználókat, hogy azok bejelentkezhessenek az AIX rendszerre.

A **SYSTEM** attribútum az /etc/security/user fájlban található. A beállítás értéke a **chuser** paranccsal módosítható:

```
#
chuser -R LDAP SYSTEM=LDAP registry=LDAP foo
```

Megjegyzés: Az ilyen típusú felügyelettel ne állítsa be alapértelmezett LDAP értékre a **SYSTEM** attribútumot. Ez lehetővé teszi az összes LDAP felhasználó számára, hogy bejelentkezzen a rendszerre.

Ez a parancs úgy állítja be az LDAP attribútumot, hogy a *foo* felhasználó számára engedélyezi a bejelentkezést erre a rendszerre. A parancs a regisztrációs adatbázist is LDAP-ra állítja, ami lehetővé teszi a *foo* bejelentkezési folyamata számára, hogy megpróbáljon bejelentkezni az LDAP-ra, ami minden felhasználókezelő feladat végrehajtását engedélyezi az LDAP-on.

Az adminisztrátornak minden egyes kliens rendszeren futtatnia kell egy ilyen beállítást ahhoz, hogy bizonyos felhasználók számára engedélyezze a bejelentkezést.

Az AIX biztosít egy olyan szolgáltatást, amely az LDAP felhasználókat úgy korlátozza, hogy csak adott LDAP kliensrendszerekbe jelentkezhetnek be. Ez a szolgáltatás központosított hoszt hozzáférés felügyeletet tesz lehetővé. Az adminisztrátor két hoszt hozzáférés felügyeleti listát adhat meg egy felhasználónak: egy engedélyező és egy tiltó listát. A rendszer ezt a két felhasználói attribútumot az LDAP szerveren tárolja a felhasználói fiókkal. A felhasználó azokhoz a rendszerekhez vagy hálózatokhoz férhet hozzá, amelyek az engedélyezési listában szerepelnek, és nem férhet hozzá azokhoz, amelyek a tiltó listában szerepelnek. Ha egy rendszer az engedélyező és tiltó listában is szerepel, akkor a felhasználó nem férhet hozzá a rendszerhez. Kétféleképpen lehet hozzáférési listákat megadni egy felhasználónak: az **mkuser** paranccsal a felhasználó létrehozásakor vagy a **chuser** paranccsal egy már meglévő felhasználónál. A lefelé kompatibilitás érdekében ha egyik lista sem létezik egy felhasználónál, akkor a felhasználó alapértelmezésben bármely LDAP kliens rendszerre bejelentkezhet.

Példák a felhasználók engedélyezési és tiltó listáinak beállításához:

```
# mkuser -R LDAP hostsallowedlogin=host1,host2 foo
```

Ez létrehoz egy *foo* felhasználót. A *foo* felhasználó csak a host1 és host2 rendszerekre jelentkezhet be.

```
# mkuser -R LDAP hostsdeniedlogin=host2 foo
```

A parancs létrehoz egy *foo* felhasználót. A *foo* felhasználó bármely LDAP kliens rendszerre bejelentkezhet, kivéve a host2 rendszert.

```
# chuser -R LDAP hostsallowedlogin=192.9.200.1 foo
```

A parancs úgy állítja be a *foo* felhasználót, hogy a felhasználó bejelentkezhessen a 192.9.200.1 címen található kliens rendszerre.

```
# chuser -R LDAP hostsallowedlogin=192.9.200/24 hostsdeniedlogin=192.9.200.1 foo
```

A parancs úgy állítja be a *foo* felhasználót, hogy a felhasználó bármely kliens rendszerre bejelentkezhet a 192.9.200/24 alhálózatban, a 192.9.200.1 címen található kliensrendszer kivételével.

További információkat a **chuser** parancsnál talál.

Biztonságos kommunikáció SSL-lel:

Az LDAP kliens és a szerver között használt hitelesítés típusától függően a jelszavak titkosított módon (unix_auth) vagy sima szövegként (ldap_auth) kerülnek elküldésre. A biztonsági kockázatok kivédésére használjon Védett socket réteget (SSL) még a titkosított jelszavak hálózaton - vagy egyes esetekben Interneten - keresztüli küldéséhez is. Az AIX tartalmaz olyan SSL csomagokat, amelyek biztonságos kommunikációs biztosítanak a címtár szerverek és a kliensek között.

További információkat a következő részben talál:

- “SSL beállítása az LDAP szerveren” oldalszám: 159
- “SSL beállítása az LDAP kliensen” oldalszám: 160

LDAPA csak hitelesítési mód használata:

Az LDAP modul teljes funkciójú modul, ami támogatja a felhasználói hitelesítést és azonosítást is. Az LDAPA modul csak hitelesítési módot biztosít. Az LDAPA modul olyan, mint az LDAP modul, de megadhatja a csak hitelesítési mód használatát.

Csak hitelesítési módban az LDAPA modul egy másik adatbázismodullal együtt működve összetett modult alkot, és nem önálló modulként működik. Az LDAPA modul felhasználói hitelesítést hajt végre, míg a második modul azonosítást végez. Ezt a kombinált modult hívják összetett modulnak. Az összetett modulhoz meg kell adni a felhasználókat az LDAP szerveren és az adatbázis szerveren is.

Az LDAPA modullal a csoport információi az adatbázis szerverről érkeznek. Például, az LDAPA fájlok esetében a csoportinformációk a helyi /etc/group fájlból származnak. Ha némely LDAP felhasználó csak LDAP csoportokhoz tartozik, akkor létre kell hozni a megfelelő LDAP csoportokat az adatbázis szerveren az LDAPA fájlok moduljának konfigurálása előtt. A megfelelő csoport létrehozásával elkerülheti, hogy az LDAPA fájlok felhasználója azért ne tudja feloldani a csoportbeállítását, mert a csoportbeállítás nem létezik az adatbázis szerveren.

Megjegyzés: Az LDAPA modul nem támogatja felhasználók létrehozását és eltávolítását. Az LDAPA fájlok felhasználójának létrehozásához a rendszeradminisztrátornak létre kell hoznia egy LDAP felhasználót az LDAP modullal, majd létre kell hozni ugyanazt a felhasználót helyileg. Ezután a felhasználót LDAPA fájlok felhasználóvá kell tennie a felhasználó SYSTEM paraméterének és a nyilvántartás beállításával LDAPAfiles értékre a **chuser** parancs segítségével.

Az LDAP konfigurálásához LDAPA modult használó csak hitelesítési módban használja az **mksecldap** parancsot az -i *<adatbázisModul>* beállítással. Ez a parancs LDAPA modult hoz létre options = authonly beállítással és LDAPA *<adatbázisModul>* összetett betöltési modullal.

Az LDAP konfigurálásához csak hitelesítési módban és a helyi fájlok adatbázismodulként használatához kövesse az alábbi példát:

```
mksecldap -c –h <ldap szerver> -a <kapcsolódásiDN> -p <kapcsolódási jelszó> -i fájlok
```

A /usr/lib/security/methods.cfg fájl a következőkkel frissül:

```
LDAPA:
        program = /usr/lib/security/LDAP
        program_64 =/usr/lib/security/LDAP64
        options = authonly

LDAP:
        program = /usr/lib/security/LDAP
        program_64 =/usr/lib/security/LDAP64

LDAPAfiles:
        options = db=BUILTIN,auth=LDAPA
```

Az LDAPA szakaszban az options = authonly beállítás jelzi, hogy az LDAPA modult csak hitelesítési módba nkell állítani. Az LDAPAfiles szakasz az összetett betöltési modult határozza meg.

Az LDAP modul megmarad a nem felhasználói/csoport adatok feloldására, mint az RBAC. Az LDAP modul továbbra is használható az LDAPA modultól független önálló hitelesítési modulként.

Kapcsolódó tájékoztatás:

mksecldap parancs

LDAPA támogatott attribútumok:

Az LDAPA modul csak hitelesítési módban korlátozott számú AIX jelszó irányelv attribútumot támogat. A többiAIX attribútumot az adatbázismodul kezeli.

A csak hitelesítési LDAPA modul a következő attribútumokat támogatja:

- maxage
- minage
- minlen
- lastupdate
- flags
- maxrepeats
- minalpha
- mindiff
- minother
- pwdwarntime
- pwdchecks
- histsize
- histexpire
- time_last_login
- time_last_unsuccessful_login
- tty_last_login
- tty_last_unsuccessful_login
- host_last_login
- host_last_unsuccessful_login
- unsuccessful_login_count
- account_locked

- loginretries
- logintimes

Nem minden LDAP szerver támogatja ezeket az attribútumokat. Ha egy LDAP szerver nem támogatja az összes felsorolt attribútumot, akkor csak a támogatott attribútumok közösek ebben a listában és a felhasználói attribútumleképezési fájlban. A leképezési fájl a /etc/security/ldap könyvtárban található.

RFC2307 szabványnak megfelelő szerver esetében AIX sématámogatás nélkül az alábbi AIX attribútumok támogatottak:

- maxage
- minage
- lastupdate
- pwdwarntime
- lastupdate

Kerberos kötés:

A kötési DN névvel és jelszóval végzett egyszerű kötés mellett a **secldapclntd** a Kerberos V hitelesítési adatok felhasználásával végzett kötést is támogatja.

A kötési azonosítóhoz tartozó kulcsok a kulcscímke fájlban tárolódnak, amelyhez a Kerberos kötés megvalósításához a **secldapclntd** a démonnak hozzáféréssel kell rendelkeznie. Ha a Kerberos kötés engedélyezett, akkor a **secldapclntd** démon a Kerberos hitelesítést a kliens /etc/security/ldap/ldap.cfg konfigurációs fájljában meghatározott azonosítónév és kulcscímke felhasználásával végzi az LDAP szerver felé. A Kerberos kötés során a **secldapclntd** démon figyelmen kívül hagyja a /etc/security/ldap/ldap.cfg fájlban meghatározott kötési megkülönböztetett nevet és jelszót.

Ha a Kerberos hitelesítés sikeres, akkor a **secldapclntd** démon elmenti a kötési hitelesítési adatokat a /etc/security/ldap/krb5cc_secldapclntd könyvtárba. Az elmentett hitelesítési adatok felhasználhatók egy esetleges későbbi kötés során. Ha azonban az ismételt kötéskor a mentett hitelesítési adatok egy óránál régebbiek, akkor a **secldapclntd** démon újból beolvassa azokat.

A Kerberos kötés használatához az **mksecldap** parancs segítségével konfigurálni kell az LDAP kliens rendszert egy kötési DN és jelszóval. Ha a konfiguráció sikeres, akkor módosítsa a /etc/security/ldap/ldap.cfg fájlban a Kerberoshoz kapcsolódó attribútumokat a megfelelő értékre. A **secldapclntd** démon újraindításkor a Kerberos kötést használja. A sikeres konfigurációt követően a démon nem használja többé a kötési megkülönböztetett nevet és jelszót. Ezeket biztonságosan el lehet távolítani a /etc/security/ldap/ldap.cfg fájlból, vagy megjegyzéssé lehet alakítani.

Kerberos azonosító létrehozása:

A Kerberos kötés támogatásához legalább két azonosítót kell létrehozni az IDS szerver számára a kulcselosztó központban (KDC). Az első az LDAP szerverhez tartozó azonosító, míg a másodikkal a kliens rendszerek végzik a kötést a szerverhez.

Mindkét azonosító kulcsot el kell helyezni egy kulcscímke fájlban, így azok felhasználhatóak a szerver folyamat illetve a kliens démon folyamatok elindításához.

Az alábbi példa az IBM Hálózati hitelesítési szolgáltatáson alapszik. Ha a Kerberos szoftvert más forrásból telepíti, akkor a használandó parancsok eltérhetnek az itt bemutatottaktól.

- Indítsa el a kadmin eszközt root felhasználóként a KDC szerveren.

  ```
  #/usr/krb5/sbin/kadmin.local
  kadmin.local:
  ```

- Hozza létre az ldap/*serverhostname* azonosítót az LDAP szerverhez. A *szerver_hosztneve* az LDAP szerver teljes képzésű DNS hosztneve.

```
kadmin.local: addprinc ldap/plankton.austin.ibm.com
WARNING: no policy specified for "ldap/plankton.austin.ibm.com@ud3a.austin.ibm.com":
Re-enter password for principal "ldap/plankton.austin.ibm.com@ud3a.austin.ibm.com":
Principal "ldap/plankton.austin.ibm.com@ud3a.austin.ibm.com" created.
kadmin.local:
```

- Hozzon létre egy kulcscímkét a létrejött azonosító számára. Ezt a kulcsot az LDAP szerver az indításkor használja. A slapd_krb5.keytab kulcscímke fájl létrehozásához adja ki a következő parancsot:

```
kadmin.local: ktadd -k /etc/security/slapd_krb5.keytab ldap/plankton.austin.ibm.com
Entry for principal ldap/plankton.austin.ibm.com with kvno 2,
encryption type Triple DES cbc mode with HMAC/sha1 added to keytab
WRFILE:/etc/security/slapd_krb5.keytab.
Entry for principal ldap/plankton.austin.ibm.com with kvno 2,
encryption type ArcFour with HMAC/md5 added to keytab WRFILE:/etc/security/slapd_krb5.keytab.
Entry for principal ldap/plankton.austin.ibm.com with kvno 2,
encryption type AES-256 CTS mode with 96-bit SHA-1 HMAC added to keytab
WRFILE:/etc/security/slapd_krb5.keytab.
Entry for principal ldap/plankton.austin.ibm.com with kvno 2,
encryption type DES cbc mode with RSA-MD5 added to keytab WRFILE:/etc/security/slapd_krb5.keytab.
kadmin.local:
```

- Az IDS adminisztrátor számára hozzon létre egy ldapadmin nevű azonosítót.

```
kadmin.local: addprinc ldapadmin
WARNING: no policy specified for ldapadmin@ud3a.austin.ibm.com; defaulting to no policy.
Note that policy may be overridden by ACL restrictions.
Enter password for principal "ldapadmin@ud3a.austin.ibm.com":
Re-enter password for principal "ldapadmin@ud3a.austin.ibm.com":
Principal "ldapadmin@ud3a.austin.ibm.com" created.
kadmin.local:
```

- Hozzon létre egy kulcscímkét a kdapadmin.keytab kötési azonosítóhoz. Ezt a **secldapclntd** kliens démon használhatja.

```
kadmin.local: ktadd -k /etc/security/ldapadmin.keytab ldapadmin
Entry for principal ldapadmin with kvno 2, encryption type
Triple DES cbc mode with HMCA/sha1 added to keytab WRFILE:/etc/security/ldapadmin.keytab.
Entry for principal ldapadmin with kvno 2, encryption type
ArcFour with HMAC/md5 added to keytab WRFILE:/etc/security/ldapadmin.keytab.
Entry for principal ldapadmin with kvno 2, encryption type
AES-256 CTS mode with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/security/ldapadmin.keytab.
Entry for principal ldapadmin with kvno 2, encryption type
DES cbc mode with RSA-MD5 added to keytab WRFILE:/etc/security/ldapadmin.keytab.
kadmin.local
```

- Hozzon létre a kliensek számára az LDAP szerverhez kapcsolódáshoz egy ldapproxy nevű azonosítót.

```
kadmin.local: addprinc ldapproxy
WARNING: no policy specified for ldapproxy @ud3a.austin.ibm.com; defaulting to no policy.
Note that policy may be overridden by ACL restriction
Enter password for principal "ldapproxy@ud3a.austin.ibm.com":
Re-enter password for principal "ldapproxy@ud3a.austin.ibm.com":
Principal "ldapproxy@ud3a.austin.ibm.com" created.
kadmin.local:
```

- Hozzon létre egy *ldapproxy.keytab* nevű kulcscímkét az **ldapproxy** kötési azonosítóhoz. Ezt a **secldapclntd** kliens démon használhatja.

```
kadmin.local: ktadd -k /etc/security/ldapproxy.keytab ldapproxy
Entry for principal ldapproxy with kvno 2, encryption type
Triple DES cbc mode with HMAC/sh1 added to keytab WRFILE:/etc/security/ldapproxy.keytab.
Entry for principal ldapproxy with kvno 2, encryption type
ArcFour with HMAC/md5 added to keytab WRFILE:/etc/security/ldapproxy.keytab
Entry for principal ldapproxy with kvno 2, encryption type
AES-256 CTS mode with 96-bit SHA-1 HMAC added to keytab WRFILE:/etc/security/ldapproxy.keytab
Entry for principal ldapproxy with kvno 2,
encryption type DES cbc mode with RSA-MD5 added to keytab WRFILE:/etc/security/ldapproxy.keytab.
kadmin.local:
```

Kerberos kötés engedélyezése az IDS szerveren:

Az alábbi eljárás engedélyezi a Kerberos kötést az IDS szerveren.

Az alábbi példa bemutatja, hogyan kell beállítani az IDS szervert a Kerberos kötéshez.

A példa IDS 5.1 változatával került tesztelésre:

1. Telepítse a krb5.client fájlkészletet.
2. Győződjön meg róla, hogy az /etc/krb5/krb5.conf fájl létezik, és helyes beállításokat tartalmaz. A konfiguráláshoz használhatja az **/usr/sbin/config.krb5** parancsot.

```
# config.krb5 -r ud3a.austin.ibm.com -d austin.ibm.com -c KDC -s alyssa.austin.ibm.com
Initializing configuration...
Creating /etc/krb5/krb5_cfg_type...
Creating /etc/krb5/krb5.conf...
The command completed successfully.
# cat /etc/krb5/krb5.conf
[libdefaults]
     default_realm = ud3a.austin.ibm.com
     default_keytab_name = FILE:/etc/krb5/krb5.keytab
     default_tkt_enctypes = des3-cbc-sha1 arcfour-hmac aes256-cts des-cbc-md5 des-cbc-crc
     defaut_tgs_enctypes = des3-cbc-shal1 arcfour-hmac aes256-cts des-cbc-md5 des-cbc-crc
[realms]
  ud3a.austin.ibm.com = {
     kdc = alyssa.austin.ibm.com:88
     admin_server = alyssa.austin.ibm.com:749
     default_domain = austin.ibm.com
  }

[domain_realm]
     .austin.ibm.com = ud3a.austin.ibm.com
     alyssa.austin.ibm.com = ud3a.austin.ibm.com

[logging]
     kdc = FILE:/var/krb5/log/krb5
     admin_server = FILE:/var/krb5/log/kadmin.log
     default = FILE:/var/krb5/log/krb5lib.log
```

3. Szerezze be az ldap:/*szerver_hosztneve* azonosítóhoz tartozó kulcscímkefájlt, és másolja az /usr/ldap/etc könyvtárba. Például: /usr/ldap/etc/slapd_krb5.keytab.
4. Állítsa be a fájl engedélyeit úgy, hogy a szerverfolyamat jogosult legyen a fájl elérésére.

```
# chown ldap:ldap/usr/ldap/etc/slapd_krb5.keytab
#
```

5. Az IDS szerveren szúrja be az alábbi bejegyzést az /etc/ibmslapd.conf fájlba a Kerberod kötés engedélyezéséhez:

```
dn: cn=Kerberos, cn-Configuration
cn: Kerberos
ibm-slapdKrbAdminDN: ldapadmin
ibm-slapdKrbEnable: true
ibm-slapdKrbIdentityMap: true
ibm-slapdKrbKeyTab: /usr/ldap/etc/slapd_krb5.keytab
ibm-slapdKrbRealm: ud3a.austin.ibm.com
objectclass: ibm-slapdKerberos
objectclass: ibm-slapdconfigEntry
objectclass: top
```

6. Képezze le a ldapproxy azonosítót egy cn-proxyuser,cn=aixdata nevű kötési megkülönböztetett névre.
   a. Ha a kötési DN már létezik az IDS szerveren, akkor hozzon létre egy ldapproxy.ldif nevű fájlt az alábbi tartalommal:

```
dn: cn=proxyuser,cn=aixdata
changetype: modify
add: objectclass
objectclass: ibm-securityidentities
```

```
-
add:altsecurityidentities
altsecurityidentities: Kerberos:ldapproxy@ud3a.austin.ibm.com
```

VAGY

b. Ha a szerver még nem tartalmazza a kötési DN-t, akkor hozzon létre egy proxyuser.ldif nevű fájlt az alábbi tartalommal:

Megjegyzés: A *proxyjelszó* helyére írja a saját jelszavát.

```
dn: cn=proxyuser,cn=mytest
cn: proxyuser
sn: proxyuser
userpassword: proxyuserpwd
objectclass: person
objectclass: top
objectclass: ibm-securityidentities
altsecurityidentities: Kerberos:ldapproxy@ud3a.austin.ibm.com
```

Az **ldapmodify** paranccsal adja hozza a létrejött kötési DN bejegyzést az IDS szerverhez.

```
# ldapmodify -D cn-admin -w adminPwd -f /tmp/proxyuser.ldif modifying entry cn=proxyuser,cn=mytest
#
```

7. Indítsa újra az IDS szervert.

Kerberos kötés engedélyezése AIX LDAP kliensen:

Az AIX LDAP kliens rendszert beállíthatja úgy, hogy az LDAP szerverhez végzett kezdeti kapcsolódáshoz Kerberost használjon.

Az IDS szervert tehát úgy kell beállítani, hogy saját maga kliense legyen.

A példa az IDS 5.1 változatával került tesztelésre:

1. Telepítse a krb5.client fájlkészletet.
2. Győződjön meg róla, hogy az /etc/krb.conf fájl létezik, és helyes beállításokat tartalmaz. Ha nincs megfelelően konfigurálva, akkor futtassa a **/usr/sbin/config.krb5** parancsot.
3. Szerezze be a kötési azonosítóhoz tartozó kulcscímkét, és másolja az /etc/security/ldap könyvtárba.
4. A fájl engedélyét állítsa 600-ra.
5. Állítsa be a klienst a **mksecldap** paranccsal a kötési DN és jelszó felhasználásával. Győződjön meg róla, hogy az AIX parancsok megfelelően működnek az LDAP felhasználókon.
6. Az /etc/security/ldap/ldap.cfg fájlban állítsa be a Kerberos rendszerrel kapcsolatos attribútumokat. Az alábbi példában a kötési azonosító az **ldapproxy** , a kulcscímke fájl pedig az **ldapproxy.keytab**. Ha IDS szerver adminisztrátori jogosultságokat szeretne, akkor cserélje le az *ldapproxy* felhasználót az *ldapadmin* felhasználóra, és az *ldapproxy.keytab* fájlt az *ldapadmin.keytab* fájlra.

   ```
   useKRB5:yes
   krbprincipal:ldapproxy
   krbkeypath:/etc/security/ldap/ldapproxy.keytab
   krbcmddir:/usr/krb5/bin/
   ```

   Ezután a kötési DN és jelszó eltávolítható az ldap.cfg fájlból (vagy megjegyzéssé alakítható), mert a **secldapclntd** démon már Kerberos kötést használ.
7. Indítsa újra a **secldapclntd** démont.
8. Az /etc/security/ldap/ldap.cfg fájlt most már terjeszteni a többi kliens rendszerre.

LDAP biztonsági információs szerver megfigyelése:

A SecureWay Directory 3.2 (és újabb) változatai alapértelmezett szerver megfigyelő naplózási funkciót biztosítanak. Miután engedélyezésre került, ez az alapértelmezett megfigyelő bedolgozó az LDAP szerver tevékenységét naplófájlba

menti. Ezzel az alapértelmezett megfigyelő bedolgozóval kapcsolatos további információkért tanulmányozza az LDAP dokumentációt a következő helyen: *Packaging Guide for LPP Installation*.

Az AIX operációs rendszerrel biztosított LDAP biztonsági információs szerver megfigyelési funkció neve: *LDAP biztonsági megfigyelő bedolgozó*. Független a SecureWay Directory alapértelmezett megfigyelő szolgáltatásától, így a két megfigyelés közül bármelyiket, vagy akár mindkettőt engedélyezni lehet. Az AIX megfigyelő bedolgozó csak azokat az eseményeket rögzíti, amelyek frissítik vagy lekérdezik egy LDAP szerver AIX biztonsági információit. Az AIX rendszer megfigyelés keretein belül működik.

LDAP-vel együttműködés érdekében a /etc/security/audit/event fájl az alábbi megfigyelési eseményeket tartalmazza:

- LDAP_Bind
- LDAP_Unbind
- LDAP_Add
- LDAP_Delete
- LDAP_Modify
- LDAP_Modifydn
- LDAP_Search

Egy ldapserver megfigyelési osztály definíció is létrehozásra kerül a fenti eseményeket tartalmazó /etc/security/audit/config fájlban.

Az LDAP biztonsági információs szerver megfigyeléséhez a /etc/security/audit/config fájl minden felhasználói szakaszához adja hozzá a következő sort:

```
ldap = ldapszerver
```

Mivel az LDAP biztonsági információs szerver megfigyelő bedolgozó az AIX rendszer megfigyelés keretein belül került megvalósításra, ezért része az AIX rendszer megfigyelő alrendszernek. Az LDAP biztonsági információs szerver megfigyelést az **audit start** és **audit shutdown** rendszer megfigyelési parancsokkal engedélyezheti illetve tilthatja le. A rendszer minden megfigyelési rekordot hozzáad a rendszer megfigyelési nyomokhoz, amelyek az **auditpr** paranccsal jeleníthetők meg. További információkat a következő rész tartalmaz: “Ellenőrzés áttekintése” oldalszám: 131.

LDAP parancsok:

Számos LDAP parancs létezik.

lsldap parancs

Az **lsldap** parancs segítségével megjeleníthetők az elnevezési szolgáltatásentitások a beállított LDAP szerverről. Ilyen entitások például: alias, automount, bootparam, ether, group, host, netgroup, network, passwd, protocol, rpc és service.

mksecldap parancs

Az **mksecldap** parancs segítségével beállíthatók az IBM SecureWay Directory szerverek és kliensek biztonsági hitelesítés és adatkezelés érdekében. A parancsot a szerveren és minden kliensen futtatni kell.

secldapclntd démon

A **secldapclntd** démon fogadja a kéréseket az LDAP betöltési modultól, továbbítja a kéréseket az LDAP biztonsági információs szerverre, és visszaadja az eredményt a szerverről az LDAP betöltési modulnak.

LDAP kezelő parancsok:

Számos parancs használható az LDAP kezelésére.

start-secldapclntd parancs

A **start-secldapclntd** parancs a **secldapclntd** démont indítja el, ha a démon még nem fut.

stop-secldapclntd parancs

A **stop-secldapclntd** parancs leállítja a futó **secldapclntd** démon folyamatot.

restart-secldapclntd parancs

A **restart-secldapclntd** parancsfájl leállítja a **secldapclntd** démont - ha az fut -, majd újraindítja. Ha a **secldapclntd** démon nem fut, akkor egyszerűen csak elindítja.

ls-secldapclntd parancs

Az **ls-secldapclntd** parancs megjeleníti a **secldapclntd** démon állapotát.

flush-secldapclntd parancs

A **flush-secldapclntd** parancs kiüríti a **secldapclntd** démon folyamat ideiglenes tárolóját.

sectoldif parancs

A **sectoldif** parancs beolvassa a helyileg definiált felhasználókat és csoportokat, és az eredményt **ldif** formátumban kinyomtatja a szabványos kimenetre.

LDAP attribútumok leképezési fájlformátuma:

Ezeket a leképezési fájlokat az /usr/lib/security/LDAP modul és a **secldapclntd** démon használja az AIX és LDAP attribútumnevek közötti leképezéshez.

A leképezési fájl minden bejegyzése egy attribútum fordítását adja meg. A bejegyzés négy, szóközökkel elválasztott mezőből áll:

```
AIX_attribútum_neve
AIX_attribútum_típusa LDAP_attribútum_neve LDAP_érték_típusa
```

A mezők leírása:

AIX_attribútum_neve
: Az AIX attribútum nevét adja meg.

AIX_attribútum_típusa
: Az AIX attribútum típusát adja meg. Az érvényes beállítások: SEC_CHAR, SEC_INT, SEC_LIST és SEC_BOOL.

LDAP_attribútum_neve
: Az LDAP attribútum nevét adja meg.

LDAP_érték_típusa
: Az LDAP érték típusát adja meg. Az érvényes típusok: **s** egy egyedülálló értéknél, és **m** a több értéknél.

LDAP és KRB5LDAP egyetlen kliensen

Ha az LDAP része egy összetett modulnak, például az KRB5LDAP modulnak, akkor csak olvasási műveletek lehetségesek, az írási műveletek nem. A **/usr/lib/security/methods.cfg** fájl alábbi konfigurációs módosításaival azonban az LDAP és az összetett betöltési modulok, például az KRB5LDAP is egyetlen fájlba kerül. Ehhez tegye a következőket:

1. Konfigurálja az LDAP klienst és a KRB5LDAP klienst a szokásos módon.
2. Módosítsa a **/usr/lib/security/methods.cfg** fájlt a következőképp:

```
LXAP:   program = /usr/lib/security/LDAP program_64
        =/usr/lib/security/LDAP64

LDAP:   program = /usr/lib/security/LDAP program_64
        =/usr/lib/security/LDAP64

NIS:    program = /usr/lib/security/NIS program_64 =
        /usr/lib/security/NIS_64

DCE:    program = /usr/lib/security/DCE

KRB5:   program = /usr/lib/security/KRB5


KRB5LXAP: options = db=LXAP,auth=KRB5
```

3. Módosítsa a **/etc/security/user** fájlban az alapértelmezett szakaszt a következőképp:

   ```
   SYSTEM = "KRB5LXAP OR LDAP OR compat"
   ```

   Az LDAP felhasználók feldolgozhatók a szokásos módon. A következő példák a KRB5LDAP felhasználók feldolgozását mutatják:

   ```
   mkuser -R KRB5LXAP <felhasználónév>
   rmuser -R KRB5LXAP <felhasználónév>
   lsuser -R KRB5LXAP <felhasználónév>
   passwd -R KRB5LXAP <felhasználónév>
   ```

EFS titkosított fájlrendszer

A titkosított fájlrendszer lehetővé teszi, hogy a rendszer egyéni felhasználói titkosítsák az adataikat a J2 fájlrendszeren az egyéni kulcstárolókon keresztül.

Minden felhasználóhoz tartozik egy kulcs. Ezek a kulcsok rejtjelezéssel védett kulcstárolóban kerülnek tárolásra és sikeres bejelentkezés esetén a felhasználó kulcsai betöltésre kerülnek a kernelbe és a rendszer hozzájuk rendelik a folyamatok hitelesítési adatait. Később, amikor a folyamatnak meg kell nyitnia egy EFS védett fájlt, akkor ezek a hitelesítési adatok tesztelésre kerülnek és ha a fájlvédelemnek megfelelő kulcs található, akkor a folyamat vissza tudja fejteni a fájlkulcsot és ezáltal a fájltartalmat is. A csoportalapú kulcskezelés szintén támogatott.

Megjegyzés: Az EFS egy átfogó biztonsági stratégia része. Úgy alakították ki, hogy együttműködjön a normális számítógépbiztonsági gyakorlatokkal és vezérlőkkel.

Titkosított fájlrendszer használhatósága

A titkosított fájlrendszer (EFS) kulcskezelése, fájltitkosítása és fájlvisszafejtése a normál műveletekben lévő felhasználók számára átlátszó.

Az EFS az alap AIX operációs rendszer része. Az EFS engedélyezéséhez a rootnak (vagy az RBAC **aix.security.efs** felhatalmazással rendelkező más felhasználónak; további információkért tekintse meg az EFS ügynökök részt) az **efsenable** paranccsal aktiválnia kell az EFS-t és létre kell hoznia az EFS környezetet. Ez egyszeri rendszerengedélyezés. Amikor az EFS engedélyezése után a felhasználó bejelentkezik, akkor a kulcsa és kulcstárolója csendesen létrehozásra kerül és védve vagy titkosításra lesz a felhasználó bejelentkezési jelszavával. A felhasználói kulcsokat ezután a J2 fájlrendszer használja csendesen az EFS fájlok titkosításakor vagy visszafejtésekor. Minden EFS fájl saját egyedi fájlkulccsal védett, és ez a fájl védett vagy titkosított lesz a fájl tulajdonosának vagy csoportjának kulcsával, a fájljogosultságoktól függően.

Alapértelmezésben a J2 fájlrendszer nem EFS-re felkészített. Ha ez EFS-re felkészített, akkor a J2 fájlrendszer átlátszó módon kezeli a titkosítást és visszafejtést a kernelben az olvasási és írási kérésekre vonatkozóan. A felhasználók és csoportok adminisztrációs parancsai (mint például az **mkgroup**, **chuser** és **chgroup**) átlátszó módon kezelik a felhasználó és csoport kulcstárolóit.

A következő EFS parancsok biztosítottak annak érdekében, hogy a felhasználó kezelhesse a kulcsokat és a fájltitkosítást:

efskeymgr
Kezeli és felügyeli a kulcsokat

efsmgr Kezeli a fájlok/könyvtárak/fájlrendszer titkosítását

Titkosított fájlrendszer ügynökök

Három típusú felhasználó kezelheti és használhatja az EFS kulcsokat:

Teljes vagy korlátozott hozzáférés rootként:

A kulcsok root hozzáférése korlátlan vagy korlátozott lehet. Egyik módban sem lehetséges, hogy a root **su** paranccsal egyszerűen átlépjen a felhasználó fiókjába és hozzáférjen a felhasználó titkosított fájljához vagy kulcstárolójához.

Az egyik módban a root vissza tudja állítani a felhasználó kulcstároló-jelszavát, és hozzáférést szerezhet a kulcstárolóban lévő felhasználói kulcsokhoz. Ez a mód nagyobb rendszeradminisztrációs rugalmasságot igényel.

A másik módban a root vissza tudja állítani a felhasználó bejelentkezési jelszavát, de a felhasználó kulcstároló-jelszavát nem. A root nem helyettesítheti a felhasználót (a **su** paranccsal) és nem örökölhet nyitott kulcstárolót. A root felhasználókat és csoportokat hozhat létre, illetve törölhet a társított társított kulcstárolókkal együtt, de nem kaphat hozzáférést a kulcstárolókban lévő kulcsokhoz. Ez a mód nagyobb mértékű védelmet biztosít a rosszindulató root támadás ellen.

Két mód áll rendelkezésre a kulcstárolók kezeléséhez és használatához: Root admin és Root védelem. Egy EFS adminisztrációs kulcs is biztosított.

Az EFS adminisztrációs kulcs lehetővé teszi az összes kulcstároló többi jelszavának elérését Root admin módban. Ez a kulcs az **efs_admin** speciális kulcstárolóban található. Az **efs_admin** speciális kulcstároló hozzáférése csak a jogosult felhasználók számára biztosított (root felhasználó és biztonsági csoport telepítéskor, vagy az RBAC **aix.security.efs** felhatalmazás).

Ha egy kulcstároló Root védelem módban van, akkor a kulcstárolóban lévő kulcsok nem kérhetők le az aktuális kulcstároló-jelszó nélkül. Ez erős biztonságot kínál a rosszindulató root ellen, de problémákat is okozhat, ha egy felhasználó elfelejti a jelszavát, mivel nincs mód a jelszó újbóli előállítására a kulcstárolóban lévő kulcsok elvesztése nélkül, és ennek eredményeképp a felhasználó a továbbiakban nem férhet hozzá az adataihoz. Ebben a kulcstároló módban néhány művelet nem kezelhető azonnal és ütemezésre kerül függőben lévő műveletként. Ezek a függőben lévő műveletek olyan esetekben kerülnek előállításra, mint például csoporthozzáférési kulcs hozzáadása vagy kikapcsolása egy felhasználói kulcstárolóban vagy magánkulcs újbóli előállítása. Ezeket a kulcstároló tulajdonosa kezeli.

efs_admin adminisztrátori kulcs:

Az efs_admin kulcstároló egy speciális kulcsot tartalmaz, amely tetszőleges felhasználói vagy csoport kulcstárolót meg tud nyitni root admin módban (az alapértelmezett mód).

A speciális kulcstároló megnyitására szolgáló jelszó a root felhasználó és biztonsági csoport kulcstárolójában kerül tárolásra EFS aktiválása esetén. Ez a jelszó hozzáadható más csoportokhoz és felhasználókhoz, vagy eltávolítható az **efskeymgr** parancs segítségével. Ez a kulcs, az RBAC **aix.security.efs** felhatalmazással együtt, lehetővé teszi, hogy a felhasználó az EFS-t adminisztrálja (azaz hozzáférjen a kulcstárolókhoz root admin módban).

efs_admin RBAC szempontok

Engedélyezett szerep alapú hozzáférés-felügyelettel rendelkező rendszereken az **efs_admin** parancs **aix.security.efs** felhatalmazással van védve.

Felhasználói kulcstárolók:

A felhasználói kulcstárolót a legtöbb általános művelet esetén a rendszer automatikusan kezeli. Az **efskeymgr** parancs karbantartási feladatok és speciális EFS használata esetén kerül felhasználásra. A felhasználók létre tudnak hozni

titkosított fájlokat és könyvtárakat az **efsmgr** parancs segítségével. A kulcstároló-kezelés a legtöbb felhasználói admin parancsba integrálva van. Felhasználó csoporthoz adásakor a felhasználó automatikusan hozzáférést kap a csoport kulcstárolójához.

A fájlhoz EFS hozzáféréssel rendelkező fájltulajdonos az **efsmgr** paranccsal EFS hozzáférést adhat más felhasználóknak és csoportoknak (a vezérléshez hasonlóan, amellyel a fájltulajdonos az ACL-ekhez rendelkezik UNIX rendszeren). A felhasználók anélkül módosíthatják jelszavukat, hogy az az azonos UID alatt megnyitott kulcstárolóval futó különálló folyamatokra hatással lenne.

Titkosított fájlrendszer kulcstároló

A kulcstárolók jelszóval védettek. A felhasználók választhatnak a bejelentkezési jelszótól különböző alternatív kulcstároló-jelszót. Ebben az esetben a kulcstároló nincs megnyitva és nem áll rendelkezésre a felhasználó szabványos bejelentkezése során. Ehelyett a felhasználónak kézzel kell betöltenie a kulcstárolót az **efskey** parancs segítségével a kulcstároló-jelszó biztosítása érdekében.

A kulcstároló-formátum **PKCS # 12**. A kulcstárolók a következő fájlokban kerülnek tárolásra:

felhasználói kulcstároló
/var/efs/users//keystore

csoport-kulcstároló
/var/efs/groups//keystore

efsadmin kulcstároló
/var/efs/efs_admin/keystore

Ha a felhasználó ugyanarra állítja be a bejelentkezési és kulcstároló-jelszavát, akkor a kulcstároló megnyitásra kerül és engedélyezett bejelentkezéskor.

A felhasználó az EFS **efskeymgr** parancs segítségével kiválaszthatja a titkosítási algoritmus típusát és a kulcshosszt.

A kulcstároló elérését a leszármazott folyamatok öröklik.

A csoportalapú kulcskezelés szintén támogatott. Csak a csoporttagok tudnak kulcsot hozzáadni a tag kulcstárolóihoz vagy eltávolítani onnan, ha a csoport-kulcstároló védelem módban van. A felhasználói kulcstároló tartalmazza a felhasználó magánkulcsát és a jelszót a felhasználó csoport-kulcstárolóinak megnyitásához, amelyek a csoport magánkulcsait tartalmazzák.

Megjegyzés: Az EFS kulcstároló csak akkor kerül automatikusan megnyitásra a szabványos AIX bejelentkezés részeként, ha a felhasználó kulcstároló-jelszava megfelel a bejelentkezési jelszónak. Ez alapértelmezésben beállításra kerül a felhasználó kulcstárolójának kezdeti létrehozása során. Elképzelhető, hogy a szabványos AIX bejelentkezéstől elérő bejelentkezési metódusok, mint például a betölthető hitelesítési modulok és cserélhető hitelesítési modulok, nem nyitják meg automatikusan a kulcstárolót.

Titkosítás és öröklés

Az EFS a J2 szolgáltatása. A fájlrendszer **efs** beállítását **igen** értékre kell álltani (tekintse meg az **mkfs** és **chfs** parancsot).

A J2 EFS automatikusan titkosítja és visszafejti a felhasználói adatokat. Ha egy felhasználó olvasás hozzáféréssel rendelkezik az EFS aktivált fájlhoz, de nem rendelkezik megfelelő kulccsal, akkor a felhasználó nem tudja normál módon olvasni a fájlt. Ha a felhasználó nem rendelkezik érvényes kulccsal, akkor az adatok nem fejthetők vissza.

Minden kriptográfiai funkció a CLiC kernel szolgáltatásból és a CLiC felhasználói függvénytárból származik.

Alapértelmezésben a J2 fájlrendszer nem EFS-re felkészített. A J2 fájlrendszernek EFS-re felkészítettnek kell lennie a fájlrendszer EFS öröklés aktiválása vagy az adatok EFS titkosítása előtt. A fájl titkosított fájlként jön létre explicit módon az **efsmgr** paranccsal vagy implicit módon az EFS öröklésen keresztül. Az EFS öröklés fájlrendszerszinten, könyvtárszinten vagy mindenkettőn aktiválható.

Az **ls** parancs felsorolja a titkosított fájl bejegyzéseit megelőző **e** karakterrel.

A **cp** és **mv** parancs a metaadatokat és a titkosított adatokat zökkenőmentesen tudja kezelni az EFS-EFS és EFS-nem EFS példahelyzetekben.

A **backup**, **restore** és **tar** parancs valamint a kapcsolódó parancsok képesek a titkosított adatok mentésére és visszaállítására, a titkosításhoz és visszafejtéshez használt EFS metaadatokat is beleértve.

Mentés és visszaállítás

Fontos az archivált EFS fájlokhoz tartozó kulcstárolók megfelelő archiválása vagy biztonsági mentése. Az archivált vagy mentett kulcstárolóhoz tartozó jelszavakat is kezelni kell és fenn kell tartani. Ezen feladatok bármelyikének elmulasztása adatvesztés eredményezhet.

EFS titkosított fájlok biztonsági mentésekor a **–Z** paraméterrel kiadott **backup** paranccsal elmenthető a fájl titkosított formája a fájl kriptográfiai metaadataival együtt. A fájladatok és metaadatok egyaránt erős titkosítással vannak védve. Ennek megvan az a biztonsági előnye, hogy a mentett fájlt erős titkosítás védi. A mentei kívánt fájlhoz tartozó fájltulajdonos és csoport kulcstárolójáról biztonsági mentést kell készíteni. Ezek a kulcstárolók a következő fájlokban találhatók:

felhasználói kulcstárolók
/var/efs/users/*user_login*/*

csoportkulcstároló
/var/efs/groups//keystore

efsadmin kulcstároló
/var/efs/efs_admin/keystore

A **restore** parancs segítségével visszaállíthat egy EFS biztonsági mentést (amely a **backup** paranccsal és a **–Z** paraméterrel lett létrehozva). A restore parancs biztosítja, hogy a kriptográfiai metaadatok is visszaállításra kerüljenek. A visszaállítási folyamat során vissza kell állítani a mentett kulcstárolókat, ha a felhasználó nem módosította az egyedi kulcstárolójában lévő kulcsokat. Amikor a felhasználó módosítja a jelszavát a kulcstároló megnyitásához, a kulcstároló belső kulcsa nem módosul. Az **efskeymgr** parancs segítségével módosíthatja a kulcstároló belső kulcsait.

Ha a felhasználó belső kulcstároló kulcsa ugyanaz marad, akkor a felhasználó azonnal megnyithatja és visszafejtheti a visszaállított fájlt az aktuális kulcstárolóval. Ha azonban a felhasználó kulcstárolójának belső kulcsa változik, akkor a felhasználónak meg kell nyitnia a mentett fájllal együtt mentett kulcstárolót. Ez a kulcstároló az **efskeymgr –o** paranccsal nyitható meg. Az **efskeymgr** parancs bekéri a felhasználó jelszavát a kulcstároló megnyitásához. Ez a kulcstárolóhoz a mentéskor használt jelszó.

Tételezzük fel például, hogy a Bob nevű felhasználó kulcstárolója **foo** jelszóval lett mentve (a ‘foo’ jelszó nem biztonságos jelszó és csak ebben a példában kerül felhasználásra az egyszerűség kedvéért), és Bob titkosított fájljainak mentése, Bob kulcstárolójával együtt, januárban került végrehajtásra. Ebben a példában Bob szintén a **foo** jelszót használja az AIX bejelentkezési jelszóhoz. Februárban Bob módosította a jelszavát **bar**-ra, amelynek hatására a kulcstároló-elérési jelszó is **bar** lett. Ha márciusban Bob EFS fájljai visszaállításra kerültek, akkor Bob az aktuális kulcstároló-jelszóval meg tudja nyitni és jeleníteni ezeket a fájlokat, mivel a kulcstároló belső kulcsát nem módosította.

Ha azonban Bob kulcstárolójának belső kulcsát módosítani kell (az **efskeymgr** paranccsal), akkor alapértelmezés szerint a régi kulcstároló belső kulcsa elavult és megmarad Bob kulcstárolójában. Amikor a felhasználó eléri a fájlt, akkor az EFS automatikusan felismeri, hogy a visszaállított fájl a régi belső kulcsot használja, és az EFS az elavult kulcsot fogja használni ennek visszafejtéséhez. Ugyanazon hozzáférés során az EFS átalakítja a fájlt az új belső kulccsal. A folyamat teljesítményére ez nincs jelentős hatással, mivel ennek kezelése a kulcstárolón és a fájl metaadatain keresztül történik, illetve nem igényli a fájladatok újbóli titkosítását.

Ha az elavult belső kulcs az **efskeymgr** paranccsal eltávolításra kerül, akkor a régi belső kulcsot tároló régi kulcstárolót vissza kell állítani és ezzel a belső kulccsal titkosított fájlokkal együtt kell használni.

Ez felveti azt a kérdést, hogy a régi jelszavak hogyan tarthatók fenn és archiválhatók biztonságosan. A jelszavak archiválásához metódusok és eszközök állnak rendelkezésre. Általában ezek a metódusok magukban foglalják a régi jelszavak listáját tartalmazó fájl tárolását, majd a fájl titkosítását és védelmét az aktuális kulcstárolóval, amely az aktuális jelszavakkal vannak védve. Az IT környezetek és biztonsági irányelvek szervezetenként különböznek, és a szervezet specifikus biztonsági igényeihez szempontokat és megfontolásokat kell megadni a környezetnek legjobban megfelelő biztonsági irányelv és gyakorlatok fejlesztése érdekében.

J2 EFS belső mechanizmus

Minden J2 EFS aktivált fájl egy speciális kiterjesztett attribútumhoz van rendelve, amely a kriptográfiai jogosultság érvényesítéséhez használt EFS metaadatokat, valamint a fájlok titkosításához és visszafejtéséhez használt információkat tartalmaz (kulcsok, kriptográfiai algoritmus, stb).

Az EA tartalom a J2 számára átlátszatlan. A felhasználói hitelesítési adatok és az EFS metaadatok is szükségesek a kriptográfiai jogosultság (hozzáférés-felügyelet) meghatározásához adott EFS aktivált fájlokhoz.

Megjegyzés: Különleges figyelmet kell fordítani azokra a helyzetekre, amelyekben fájl vagy adat elveszhet (például a fájl EA eltávolítása).

EFS védelemöröklés

Az EFS aktiválás után az újonnan létrehozott közvetlen leszármazottak automatikusan EFS aktiváltak lesznek, amennyiben ez nem kerül kézzel felülbírálásra. A szülőkönyvtár EFS attribútumai elsőbbséget élveznek a fájlrendszer EFS attribútumaival szemben.

Egy könyvtár öröklési hatóköre pontosan egy szint. Az újonnan létrehozott leszármazottak öröklik a szülő EFS attribútumait, amennyiben a szülőkönyvtár EFS aktivált. A meglévő leszármazott fenntartja az aktuális titkosított vagy nem titkosított állapotot. A logikai öröklési lánc megszakad, ha a szülő módosítja az EFS attribútumait. A módosítások nem terjednek a könyvtár meglévő leszármazottai felé, így azokra a könyvtárakra kézzel kell alkalmazni.

Munkapartíció-szempontok

Mielőtt a titkosított fájlrendszert engedélyezné vagy használná a munkapartícióban, az EFS-et először a globális rendszeren kell engedélyezni az **efsenable** paranccsal. Ezt az engedélyezést csak egyszer kell végrehajtani. Ezen felül minden fájlrendszert, az EFS-re felkészítettet is beleértve, a globális rendszerről kell létrehozni.

Titkosított fájlrendszer beállítása

Ezt először kell végrehajtani.

A szakaszt így kell beállítani.

1. Telepítse a **clic.rte** fájlkészletet. A fájlkészlet tartalmazza az EFS által igényelt kriptográfiai könyvtárakat és kernelkiterjesztést. A **clic.rte** fájlkészlet az AIX bővítőcsomagban található.
2. Engedélyezze az EFS-t a rendszeren az **efsenable** paranccsal (például: >efsenable –a). Jelszó kérése esetén érdemes a root jelszót használni. A felhasználók kulcstárolói automatikusan létrehozásra kerülnek, majd a felhasználó bejelentkezik, vagy újra bejelentkezik az **efsenable** parancs futtatása után. Az **efsenable –a** lefuttatása után a rendszer EFS-re felkészített lesz és az **efsenable** parancsot nem kell újra futtatni.
3. Hozzon létre egy EFS-re felkészített fájlrendszert az **–a efs=yes** paraméterrel. Például: crfs -v jfs2 -m /foo –A yes -a efs=yes -g rootvg -a size=20000
4. A fájlrendszer felépítése után kapcsolja be a kriptográfiai öröklést az EFS-re felkészített fájlrendszeren. Ez az **efsmgr** paranccsal hajtható végre. Az előző példa folytatásához, amelyben a **/foo** fájlrendszer létrehozásra került, futtassa a következő parancsot: efsmgr –s –E /foo. Ez lehetővé teszi, hogy a fájlrendszeren létrehozott és használt összes fájl titkosított legyen.

Ettől a ponttól fogva, amikor a felhasználó vagy folyamat egy megnyitott kulcstárolóval fájlt hoz létre a fájlrendszeren, akkor a fájl titkosításra kerül. Amikor a felhasználó vagy fájl olvassa a fájlt, a fájl automatikusan visszafejtésre került azon felhasználók számára, akik jogosultak a fájl hozzáférésére.

További információkért tekintse meg a következőket:

- **chfs**, **chgroup**, **chuser**, **cp**, **efsenable**, **efskeymgr**, **efsmgr**, **lsuser**, **ls**, **mkgroup**, **mkuser** és **mv** parancs
- /etc/security/group és /etc/security/user fájl

Titkosított fájlrendszerek kulcstárolóinak távoli elérése

Vállalati környezetben központosíthatja a titkosított fájlrendszer (EFS) kulcstárolóit. Ha a kulcstárolókat vezérlő adatbázisokat az egyes rendszereken függetlenül tárolja, akkor nehéz lehet a kulcstárolók kezelése. Az AIX központosított EFS kulcstároló lehetővé teszi a felhasználó és csoport kulcstároló adatbázisok tárolását LDAP címtárban, hogy központilag kezelhesse az EFS kulcstárolót.

Kapcsolódó fogalmak:

“Egyszerűsített címtárhozzáférési protokoll” oldalszám: 150
Az Egyszerűsített címtárhozzáférési protokoll (LDAP) a címtárak (vagy adatbázisok) információinak helyi vagy távoli hozzáférésének illetve frissítésének szabványos módszerét adja meg egy kliens-szerver modellben.

Áttekintés a titkosított fájlrendszerek kulcstárolóinak távoli eléréséről:

Információkat találhat a titkosított fájlrendszerek (EFS) adatbázisairól, az EFS parancsok LDAP engedélyezéséről és az egyedi kulcstároló hozzáférésről.

Minden AIX EFS kulcstároló adatbázist tárolhat LDAP címtárban, amely tartalmazza a következő EFS adatbázisokat:

- Felhasználói kulcstároló
- Csoport kulcstároló
- Admin kulcstároló
- Cookie-k

Az AIX operációs rendszer segédprogramokat biztosít, melyek segítséget nyújtanak az alábbi felügyeleti feladatok végrehajtásában:

- Helyi kulcstároló adatok exportálása LDAP szerverre
- A kliens konfigurálása LDAP szerveren tárolt EFS kulcstároló adatok használatára
- EFS kulcstároló adatok hozzáférésének szabályzása
- LDAP adatok kezelése kliensrendszerről

Minden EFS kulcstároló adatbázis-kezelő parancs engedélyezve van az LDAP kulcstároló adatbázis használatára. Ha a rendszerszintű keresési sorrend nincs megadva a /etc/nscontrol.conf fájlban, akkor a kulcstároló műveletek a felhasználói és csoport efs_keystore_access attribútumtól függnek. Ha az efs_keystore_access értéke ldap, akkor az EFS parancsok kulcstároló műveleteket hajtanak végre az LDAP kulcstárolón.

A következő táblázat az EFS parancsok módosításait írja le az LDAP címtárhoz.

12. táblázat: EFS parancsengedélyezés LDAP címtárhoz

Parancs	LDAP információk
Bármilyen EFS parancs	Ha az efs_keystore_access attribútum értéke ldap, akkor nem kell használnia az -L *tartomány* különleges beállítást semmilyen paranccsal, hogy kulcstároló műveleteket hajtson végre az LDAP címtáron.
efskeymgr	Tartalmazza az -L load_module beállítást, hogy kifejezett kulcstároló műveleteket hajthasson végre az LDAP címtáron.
efsenable	Tartalmazza a -d Basedn beállítást, hogy végrehajthassa az LDAP kezdeti beállítását az EFS kulcstároló befogadásához. A kezdeti beállítás tartalmazza az EFS kulcstároló alap megkülönböztetett neveinek (DN) hozzáadását és a helyi könyvtárstruktúra (/var/efs/) létrehozását.

12. táblázat: EFS parancsengedélyezés LDAP címtárhoz (Folytatás)

Parancs	LDAP információk
efskstoldif	Létrehozza az LDAP számára az EFS kulcstároló adatokat a helyi rendszer következő adatbázisaiból: • /var/efs/users/*felhasználónév*/keystore • /var/efs/groups/*csoportnév*/keystore • /var/efs/efs_admin/keystore • Cookie-k, ha léteznek, az összes kulcstárolónál

Minden kulcstároló bejegyzésnek egyedinek kell lennie. Minden kulcstároló bejegyzés közvetlenül megfelel a felhasználó- és csoportnevet tartalmazó bejegyzés megkülönböztetett nevének. A rendszer lekérdezi a felhasználói azonosítókat (uidNumber), csoportazonosítókat (gidNumber) és a megkülönböztetett neveket. A lekérdezés akkor sikeres, ha a felhasználó és csoportnevek egyeznek a nekik megfelelő megkülönböztető nevekkel. Mielőtt létrehozná vagy áttelepítené az EFS kulcstároló bejegyzéseket az LDAP szerveren, gondoskodjon arról, hogy a felhasználó- és csoportnevek és azonosítók egyediek legyenek a rendszeren.

Kapcsolódó feladatok:

“Titkosított fájlrendszer kulcstároló adatainak exportálása LDAP címtárba”
Fel kell töltenie az LDAP szervert a kulcstároló adataival, hogy az LDAP címtárt a titkosított fájlrendszer (EFS) kulcstárolójának központosított lerakataként használhassa.

“LDAP kliens konfigurálása titkosított fájlrendszer kulcstárolójához”
LDAP címtárban tárolt titkosított fájlrendszer (EFS) kulcstároló adatok használatához konfigurálnia kell a rendszert LDAP kliensként.

Titkosított fájlrendszer kulcstároló adatainak exportálása LDAP címtárba:

Fel kell töltenie az LDAP szervert a kulcstároló adataival, hogy az LDAP címtárt a titkosított fájlrendszer (EFS) kulcstárolójának központosított lerakataként használhassa.

Mielőtt létrehozná vagy áttelepítené az EFS kulcstároló bejegyzéseket az LDAP szerveren, gondoskodjon arról, hogy a felhasználó- és csoportnevek és azonosítók egyediek legyenek a rendszeren.

Az LDAP szerver feltöltéséhez az EFS kulcstároló adatokkal tegye a következőket:

1. Telepítse az EFS kulcstároló sémát az LDAP szerverre:
   a. Olvassa be az LDAP EFS kulcstároló sémáját az /etc/security/ldap/sec.ldif fájlból azAIX rendszeren.
   b. Az **ldapmodify** parancs futtatásával frissítse az LDAP szerver sémáját az LDAP EFS kulcstároló sémájával.
2. Az **efskstoldif** parancs futtatásával olvassa be a helyi EFS kulcstárolófájlok adatait és készítsen az LDAP számára megfelelő formátumú kimenetet. Egyedi kulcstároló hozzáférés megtartásához helyezze az LDAP címtárban található EFS kulcstároló adatokat ugyanazon szülő megkülönböztetett neve (DN) alá, mint a felhasználói és csoportadatokat.
3. Adatok mentése fájlba.
4. Futtassa az **ldapadd -b** parancsot az LDAP szerver feltöltéséhez a kulcstároló adataival.

Kapcsolódó fogalmak:

“Áttekintés a titkosított fájlrendszerek kulcstárolóinak távoli eléréséről” oldalszám: 174
Információkat találhat a titkosított fájlrendszerek (EFS) adatbázisairól, az EFS parancsok LDAP engedélyezéséről és az egyedi kulcstároló hozzáférésről.

LDAP kliens konfigurálása titkosított fájlrendszer kulcstárolójához:

LDAP címtárban tárolt titkosított fájlrendszer (EFS) kulcstároló adatok használatához konfigurálnia kell a rendszert LDAP kliensként.

LDAP kliens konfigurálásához EFS kulcstárolóhoz tegye a következőket:

1. Futtassa a **/usr/sbin/mksecldap** parancsot a rendszer konfigurálásához LDAP kliensként. A **mksecldap** parancs dinamikus keresést végez a megadott LDAP szerveren az EFS kulcstároló adatok helyének meghatározásához. Ezután elmenti az eredményeket a /etc/security/ldap/ldap.cfg fájlba. A **mksecldap** parancs meghatározza a user, group, admin és efscookies kulcstárolók adatainak helyét.
2. Tegye a következők egyikét az LDAP engedélyezéséhez az EFS kulcstároló adatok kikeresési tartományaként:
   - Állítsa be a felhasználói és csoport **efs_keystore_access** attribútumot **file** vagy **ldap** értékre.
   - Adja meg a kulcstároló keresési sorrendjét rendszerszinten a /etc/nscontrol.conf fájl segítségével. A következő táblázat egy példát tartalmaz.

13. táblázat: A /etc/nscontrol.conf fájl példakonfigurációja

Attribútum	Leírás	Keresési sorrend (secorder)
efsusrkeystore	Ez a keresési sorrend minden felhasználónál közös.	LDAP, files
efsgrpkeystore	Ez a keresési sorrend minden csoportnál közös.	files, LDAP
efsadmkeystore	Ez a keresési sorrend a cél kulcstárolók admin kulcstárolóját keresi meg.	LDAP, files

> **FIGYELEM:** A /etc/nscontrol.conf fájlban megadott konfiguráció felülbírálja a felhasználói és csoport **efs_keystore_access** attribútumhoz megadott értékeket. Ugyanez igaz a felhasználói **efs_adminks_access** attribútumra is.

Miután konfigurálta a rendszert LDAP kliensként és engedélyezte az LDAP szervert az EFS kulcstároló adatok kikeresési tartományaként, a /usr/sbin/secldapclntd kliensdémon beolvassa az EFS kulcstároló adatokat az LDAP szerverről, amikor LDAP kulcstároló műveleteket hajt végre.

Kapcsolódó fogalmak:

“Áttekintés a titkosított fájlrendszerek kulcstárolóinak távoli eléréséről” oldalszám: 174
Információkat találhat a titkosított fájlrendszerek (EFS) adatbázisairól, az EFS parancsok LDAP engedélyezéséről és az egyedi kulcstároló hozzáférésről.

Nyilvános kulcsú titkosítási szabványok #11

A Nyilvános kulcsú titkosítási szabványok (PKCS #11) alrendszer lehetővé teszi az alkalmazások számára, hogy a hardvereszközöket (tokeneket) az eszköz típusától függetlenül érjék el.

Az alábbi fejezet a PKCS #11 szabvány 2.20 verziójának felel meg.

A PKCS #11 alrendszer a következő összetevőket használja:

- Egy API megosztott objektum (/usr/lib/pkcs11/ibm_pks11.so), amely általános felületként szolgál a PKCS #11 szabványt támogató eszközillesztőkhöz. Ez a réteges felépítés lehetővé teszi az új PKCS #11 eszközök használatát anélkül, hogy újra kelljen fordítania a meglévő alkalmazásokat.
- Egy PKCS #11 eszközillesztő, amely az alkalmazások számára hasonló képességeket biztosít, mint más kernel összetevők számára, mint például a titkosított fájlrendszer (EFS) vagy az IP biztonság (IPSec).
- Ha a platform támogatja a kriptográfia társprocesszort, a PKCS #11 eszközillesztő a Fejlett titkosítási szabvány (AES), a Biztonságos kivonatkészítési algoritmus (SHA) és a Kivonatüzenet hitelesítési kód (HMAC) műveletekhez elérhető hardveres gyorsítást használja. A jobb teljesítmény érdekében engedélyezheti a hálózati memóriakapcsolatot.

Kapcsolódó tájékoztatás:

AIX memóriakapcsolat támogatása

IBM 4758 Model 2 kriptográfiai társprocesszor

Az IBM 4758 Model 2 kriptográfiai társprocesszor biztonságos feldolgozási környezetet biztosít.

Mielőtt megkísérelné a PKCS #11 alrendszer beállítását, ellenőrizze, hogy az adapter helyesen lett-e beállítva egy támogatott mikrokóddal.

IBM 4960 Cryptographic Accelerator

Az IBM 4960 Cryptographic Accelerator eszközt kínál a titkosítási tranzakciók kitöltéséhez. Mielőtt megkísérelné a PKCS #11 alrendszer beállítását, ellenőrizze, hogy az adapter helyesen lett-e beállítva.

Az IBM 4758 Model 2 kriptográfiai társprocesszor ellenőrzése Nyilvános kulcsú kriptográfiai szabványok #11 alrendszerrel való használatra:

A PKCS #11 alrendszer úgy készült, hogy telepítéskor és újraindításkor automatikusan felderítse a PKCS #11 hívásokat kiszolgálni képes kártyákat. Emiatt azokat az IBM 4758 Model 2 kriptográfiai társprocesszorokat, amelyek nincsenek helyesen beállítva, a PKCS #11 illesztő nem fogja elérni és az adapternek küldött hívások meghiúsulnak.

Az adapter beállításának ellenőrzéséhez tegye a következőket:

1. Az alábbi paranccsal ellenőrizze, hogy az adapter szoftvere megfelelően lett-e telepítve:
   ```
   lsdev -Cc adapter | grep crypt
   ```

   Ha az IBM 4758 Model 2 kriptográfiai társprocesszor nem jelenik meg az eredményül kapott listában, akkor ellenőrizze, hogy a kártya megfelelően van-e behelyezve, illetve hogy a hozzá tartozó szoftver telepítve van-e.
2. A kártya firmware helyességének meghatározásához írja be a következő parancsot:
   ```
   csufclu /tmp/l ST device_number_minor
   ```

   Ellenőrizze, hogy a Segment 3 képfájlba a PKCS #11 alkalmazás be van-e töltve. Ha nincs betöltve, forduljon a kártya dokumentációjához, hogyan szerezheti be a legfrissebb mikrokódot, illetve hogyan kell azt telepítenie.

   Megjegyzés: Ha ez a segédprogram nem áll rendelkezésre, akkor a támogató szoftver nem került telepítésre.

Az IBM 4960 Model 2 Cryptographic Accelerator ellenőrzése Nyilvános kulcsú kriptográfiai szabványok #11 alrendszerrel való használatra:

A PKCS #11 alrendszer úgy készült, hogy telepítéskor és újraindításkor automatikusan felderítse a PKCS #11 hívásokat kiszolgálni képes kártyákat. Emiatt azokat az IBM 4960 Cryptographic Acceleratorokat, amelyek nincsenek megfelelően beállítva, a PKCS #11 illesztő nem fogja elérni és az adapternek küldött hívások meghiúsulnak.

Az alábbi paranccsal ellenőrizze, hogy a kártya szoftvere megfelelően lett-e telepítve:

```
lsdev -Cc adapter | grep ica
```

Ha az IBM 4960 Cryptographic Accelerator nem található meg az eredményül kapott listában, akkor ellenőrizze, hogy a kártya megfelelően van-e behelyezve, illetve hogy a támogató illesztőprogram megfelelően telepítve van-e.

Nyilvános kulcsú titkosítási szabványok #11 alrendszer beállítása

A PKCS #11 alrendszer automatikusan felismeri a PKCS #11-et támogató eszközöket. Ahhoz azonban, hogy egyes alkalmazások is használhassák ezeket az eszközöket, némi kezdeti beállításra szükség van.

E feladatok elvégezhetők az API-n keresztül (egy megfelelő PKCS #11 alkalmazás írásával), vagy a SMIT felületen. A PKCS #11 SMIT beállításai elérhetők egyrészt a SMIT főmenüjének **PKCS11 alrendszer kezelése** pontjából, másrészt a **smit pkcs11** gyorseléréssel.

A token inicializálása:

A kártya vagy PKCS #11 tokeneket a sikeres használathoz inicializálni kell.

Az inicializálási eljárás során a token egyedi címkét kap. Az alkalmazások ezzel a címkével azonosíthatják egyedi módon a tokent. Éppen ezért a címkék nem ismételhetők. Az API azonban nem ellenőrzi, hogy a címkék valóban nem lettek újra felhasználva. Az inicializálást elvégezheti egy PKCS #11 alkalmazás, vagy a rendszeradminisztrátor a SMIT

segítségével. Ha a tokenben van adatvédelmi megbízott PIN, akkor az alapértelmezett érték 87654321-re kerül beállításra. A PKCS #11 alrendszer biztonsága érdekében ezt az értéket inicializálás után meg kell változtatni.

A token inicializálásához:
1. Lépjen be a tokenkezelési képernyőre a smit pkcs11 gyorseléréssel.
2. Válassza ki a **Token inicializálása** pontot.
3. Válasszon ki egy PKCS #11 kártyát a támogatott kártyák megjelenő listájából.
4. Erősítse meg a választást az Enter lenyomásával.

   Megjegyzés: Ez a token minden információját kitörli.
5. Adja meg az adatvédelmi megbízott PIN-t (SO PIN) és egy egyedi tokencímkét.

A helyes PIN beírása esetén a kártya inicializálásra (vagy újrainicializálásra) kerül a parancs lefutása után.

Adatvédelmi megbízott PIN beállítása:

A SO PIN alapértelmezett értékének lecseréléséhez tegye a következőket.

A PIN alapértelmezett értékének lecserélése:
1. Írja be, hogy smit pkcs11.
2. Válassza ki az **Adatvédelmi megbízott PIN beállítása** pontot.
3. Válassza ki azt az inicializált adaptert, amelyhez be akarja állítani a PIN-t.
4. Adja meg a jelenlegi és az új PIN-t.
5. Írja be ellenőrzésképpen még egyszer az új PIN-t.

A felhasználói PIN inicializálása:

A token inicializálása után lehet, hogy be kell állítani a felhasználói PIN-t ahhoz, hogy az alkalmazások elérhessék a token objektumait.

Annak megállapításához, hogy be kell-e jelentkeznie a felhasználónak az objektumok eléréséhez, tekintse meg az eszköz dokumentációjához.

A felhasználói PIN inicializálásához:
1. Lépjen be a tokenkezelési képernyőre a smit pkcs11 gyorseléréssel.
2. Válassza ki a **Felhasználói PIN inicializálása** pontot.
3. Válasszon ki egy PKCS #11 kártyát a támogatott kártyák megjelenő listájából.
4. Adja meg az SO PIN-t és a felhasználói PIN-t.
5. Írja be újra ellenőrzésként a felhasználói PIN-t.
6. Sikeres ellenőrzés esetén a felhasználói PIN módosításra kerül.

Felhasználói PIN alaphelyzetbe állítása:

A felhasználói PIN alaphelyzetbe állításához inicializálja újra a PIN-t az SO PIN-nel, vagy állítsa be a felhasználói PIN-t a meglévő felhasználói PIN segítségével.

PIN alaphelyzetbe állítása:
1. Lépjen be a tokenkezelési képernyőre a smit pkcs11 gyorseléréssel.
2. Válassza ki a **Felhasználói PIN beállítása** pontot.
3. Válassza ki azt az inicializált adaptert, amelynek felhasználói PIN-jét be akarja állítani.
4. Adja meg a jelenlegi felhasználói PIN-t, majd az új PIN-t.

5. Írja be ellenőrzésképpen még egyszer az új PIN-t.

Nyilvános kulcsú titkosítási szabványok #11 használata

Ahhoz, hogy egy alkalmazás használhassa a PKCS #11 alrendszert, az alrendszer kártyahelykezelő démonjának futnia kell, az alkalmazásnak pedig be kell töltenie az API megosztott objektumát.

A kártyahelykezelőt általában rendszerbetöltéskor indítja el az **inittab**, a /etc/rc.pkcs11 parancsfájlt meghívásával. Ez a parancsfájl ellenőrzi a rendszer kártyáit, mielőtt elindítaná a kártyahelykezelő démont. Ez azt eredményezi, hogy a kártyahelykezelő démon nem érhető el addig, amíg a felhasználó be nem jelentkezik a rendszerbe. A démon elindulása után az alrendszer a rendszeradminisztrátor beavatkozása nélkül is feljegyzi a támogatott kártyák számának vagy típusának módosulását.

Az API betölthető akár futásidőben, az objektum csatolásával, akár késleltetett szimbólumfeloldással. Egy alkalmazás például lekérheti a PKCS #11 funkciólistát az alábbi módon:

```
d CK_RV (*pf_init)();
void *d;
CK_FUNCTION_LIST *functs;

d = dlopen(e, RTLD_NOW);
if ( d == NULL ) {
   return FALSE;
}

pfoo = (CK_RV (*)())dlsym(d, "C_GetFunctionList");
if (pfoo == NULL) {
   return FALSE;
}

rc = pf_init(&functs);
```

Nyilvános kulcs titkosítási szabványok #11 eszközök

Az AIX rendszeren belül két eszköz érhető el a titkosítási rendszerek kezelésére: a PKCS #11 kulcskezelési eszköz és a PKCS #11 adminisztrációs eszköz. Ezeket az eszközöket a Curses alapú grafikus felületről vagy a parancssorból érheti el.

Megjegyzés: Az AIX titkosítási keretrendszer eszközeihez kötegelt feldolgozási képesség szükséges. A kötegelt feldolgozási képességek részletes információiért lásd: “Kötegelt feldolgozás” oldalszám: 181.

A PKCS #11 kulcskezelési eszköz AIX rendszeren a kulcsok, igazolások és PKCS #11 adatok kezelésének központosított eszköze. Az eszköz által kezelt objektumok vagy támogatott PKCS #11 szolgáltatókon, úgymint az IBM kriptográfiai adaptercsaládon (például az IBM 4758, 4960 és 4764), vagy az AIX Titkosítási keretrendszeren belül vannak tárolva. A PKCS #11 kulcskezelési eszközzel különböző műveleteket hajthat végre. A műveletek közé tartoznak a PKCS #10 igazolás-aláírási kérés (CSR) vagy a saját aláírású igazolások előállítása. Emellett az eszköz segítségével keresheti, megtekintheti, törölheti, importálhatja, exportálhatja és elmentheti a PKCS #11 objektumadatokat, illetve átviheti a PKCS #11 objektumadatokat PKCS #11 tokenek között. Az eszköz grafikus felületét a **p11km** paranccsal indíthatja el. Az eszköz betölt minden elérhető PKCS #11 tokent. A nyíl gombok segítségével a tokenek listájában fel- és le görgetve megtekintheti a tokenek részleteit. Token kiválasztásához a nyíl gombok segítségével válassza ki a tokent, majd nyomja meg az Enter gombot. Az eszköz parancssori verzióját a következő parancs futtatásával indíthatja el:

```
p11km -b <parancsfájl>
```

A PKCS #11 adminisztrációs eszköz az AIX PKCS #11 titkosítási keretrendszer kezelésének központosított eszköze. Az eszköz segítségével az adminisztrátor vagy a biztonsági szakember kezelheti az AIX titkosítási keretrendszer által vezérelt tokeneket. Az eszköz segítségével inicializálhatja, létrehozhatja és megsemmisítheti a PKCS #11 tokeneket, kezelheti a bővítőhelyeket, visszaállíthatja a felhasználói jelszavakat, megerősítheti az objektumok törlését, megadhatja az objektumok megbízhatóságát és végrehajthatja az AIX titkosítási keretrendszer finomhangolását a teljesítmény javítása céljából. Az eszköz grafikus felületét a **p11admin** paranccsal indíthatja el. Az eszköz betölt minden elérhető PKCS #11 tokent. A nyíl gombok segítségével a tokenek listájában fel- és le görgetve megtekintheti a tokenek

részleteit. Token kiválasztásához a nyíl gombok segítségével válassza ki a tokent, majd nyomja meg az Enter gombot. Az eszköz parancssori verzióját a következő parancs futtatásával indíthatja el:

```
p11admin -b <parancsfájl>
```

Parancs profilok:

Az AIX titkosítási keretrendszer az OpenSSL függvénytárat használja az egyéni profilok létrehozásában felhasznált konfigurációs fájlok értelmezéséhez. Ezeknek a profiloknak a segítségével beállíthatja az eszköz attribútumait, például a **p11km** és a **p11admin** parancs grafikus felületének színeit.

A “Kötegelt feldolgozás” oldalszám: 181 című részben megadott fájlformátum segítségével létrehozhatja és szerkesztheti a következő profilfájlokat a grafikus felület személyre szabásához.

Megjegyzés: A profilfájlok létrehozása után a következőképp nevezze el és tárolja el a saját könyvtárában:

```
$HOME/.p11km
```

```
$HOME/.p11admin
```

A grafikus felület következő színattribútumai támogatottak:

```
action_name = "GUI_COLORS"
gui_fg_color = "<szín neve>" ## Előtér színe
gui_bg_color = "<szín neve>" ## Háttér színe
gui_vc_color = "<szín neve>" ## Megtekintett tartalom színe
```

Ahol <*szín neve*> a következő értékek egyike:

- LIGHT GRAY
- WHITE
- BLACK
- DARK GRAY
- RED
- LIGHT RED
- YELLOW
- ORANGE vagy BROWN
- GREEN
- LIGHT GREEN
- BLUE
- LIGHT BLUE
- CYAN
- LIGHT CYAN
- MAGENTA
- LIGHT MAGENTA

Példa: p11km profile ($HOME/.p11km)

```
[p11km_cmd]
gui_fg_color = "RED"
gui_bg_color = "BLACK"
gui_vc_color = "WHITE"
```

Példa: p11admin Profile ($HOME/.p11admin)

```
[p11admin_cmd]
gui_fg_color = "BLUE"
gui_bg_color = "LIGHT GRAY"
gui_vc_color = "BLACK"
```

Kötegelt feldolgozás:

A parancssorból futtathatja a kötegelt feldolgozás parancsait ugyanazon feladatok végrehajtására, amelyek a PKCS #11 eszközök grafikus felületű verzióiból elérhetők.

A PKCS #11 kulcskezelési eszköz (p11km) parancsformátuma a következő:

```
p11km -b <parancsfájl>
```

A PKCS #11 kulcsadminisztrációs eszköz (p11admin) parancsformátuma a következő:

```
p11admin -b <parancsfájl>
```

Mivel ezek az eszközök az OpenSSL könyvtár segítségével értelmezik a parancsfájlokat, a parancsfájlok formátuma követi a szokásos OpenSSL konfigurációs fájlformátumot. Minden szakasz egy külön parancs, és az attribútum/érték párok adják meg a feldolgozáshoz szükséges információkat. Minden szakaszparancs kötegelt feldolgozása felülről lefelé következik be. Ha egy egyéni kötegelt parancs meghiúsul, akkor a program hibát ír ki és a kötegelt feldolgozás az ezt követő szakaszparancsok feldolgozása nélkül befejeződik.

A következő példa az OpenSSL konfigurációs fájlformátumra.

```
[szakasz1]
attribútum1 = "érték1"
attribútum2 = "érték2"
...
attribútumN = "értékN"
[szakasz2]
attribútum1 = "érték1"
attribútum2 = "érték2"
...
attribútumN = "értékN"
...
...
[szakaszN]
attribútum1 = "érték1"
attribútum2 = "érték2"
...
attribútumN = "értékN"
```

Annak biztosítására, hogy a PKCS #11 eszköz parancsszakaszai meglegyenek az OpenSSL konfigurációs fájl szakaszai mellett, használja a következő előtagokat a PKCS #11 szakaszokhoz:

p11km eszköz
: p11km_cmd

p11admin eszköz
: p11admin_cmd

Minden p11km_cmd vagy p11admin_cmd szakasz csak egy action_name attribútumot tartalmazhat a szakasszal társított parancs azonosítására szolgáló karakterlánc értékkel. A legegyszerűbb példa egy fájl, amely egy parancsszakaszt tartalmaz, amely további paraméterekkel nem rendelkező parancsot ír le. A következő példa bemutatja, hogyan kell használni a p11km eszközt a rendszeren lévő elérhető PKCS #11 tokeneket felsoroló kötegelt parancs futtatására:

```
[p11km_cmd_list_my_tokens]
action_name="LIST_TOKENS"
```

Minden kötegelt parancs támogat egy választható logikai attribútumot:

```
start_gui="<logikai érték>"
```

Ha TRUE értékű logikai attribútumot tartalmazó kötegelt parancsot futtat, akkor a kötegelt feldolgozás a parancs befejeződése után befejeződik és elindul a grafikus felület.

Megjegyzés: Ha egy parancsfájl a választható **start_gui** attribútumot tartalmazó parancsot tartalmaz, akkor az utána felsorolt egyik kötegelt parancs feldolgozására sem kerül sor.

Kötegelt parancsok:

A kötegelt parancsok parancssori hozzáférést biztosítanak a PKCS #11 eszközökhöz.

A következő kötegelt parancsok érhetők el a PKCS #11 kulcskezelési eszközben (p11km).

Megjegyzés: A kötegelt parancsok használatához tegye a következőket:

1. Hozzon létre és szerkesszen egy parancsfájlt a következő fejezetben leírtak szerint: “Kötegelt feldolgozás” oldalszám: 181.
2. Hozzon létre új p11km_cmd szakaszokat, amelyek tartalmazzák a használni kívánt kötegelt parancsok attribútumait.

Elérhető PKCS #11 tokenek felsorolása

Jelentést hoz létre és megjeleníti az elérhető PKCS #11 tokenek token és bővítőhely információit.

Kötelező attribútumok

```
action_name = "LIST_TOKENS"
```

Választható attribútumok

```
start_gui = "<logikai érték>"
```

Ahol <*logikai érték*> értéke vagy TRUE, vagy FALSE

Példa

```
[p11km_cmd_list_tokens]
action_name = "LIST_TOKENS"
```

Elérhető PKCS#11 mechanizmusok felsorolása

Jelentést hoz létre és megjeleníti egy (az illesztőprogram és a bővítőhely attribútumértékeinek megadásával megkapott) adott PKCS #11 token által támogatott elérhető PKCS #11 mechanizmusokat.

Kötelező attribútumok

```
action_name = "LIST_MECHANISMS"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
```

Ahol <*bővítőhely száma*> pozitív egész érték, az <*illesztőprogram neve*> pedig a következő értékek egyike:

Érték	Leírás
AIX	AIX operációs rendszer titkosítási keretrendszere
IBM_4758_4960	IBM 4758/4960 titkosítási hardveradapterek
IBM_4764	IBM 4764 titkosítási hardveradapterek
Other	Ha OTHER értéket ad meg, akkor meg kell adnia a **p11_driver_path** attribútumot is.

Választható attribútumok

```
start_gui = "<logikai érték>"
```

Kiegészítő attribútumok

```
p11_driver_path = "<a PKCS#11 illesztőprogram útvonala>"
```

Ahol <*PKCS#11 illesztőprogram útvonala*> a parancshoz használt PKCS #11 függvénytár teljes UNIX elérési útja és fájlneve. Ez az attribútum csak akkor adható meg, ha a **p11_driver** attribútum értéke OTHER.

Példa

```
[p11km_cmd_list_4764_slot_0_mechs]
action_name = "LIST_MECHANISMS"
p11_driver = "IBM_4764"
p11_slot = "0"
start_gui = "TRUE"
```

Elérhető PKCS #11 objektumok felsorolása

Jelentést hoz létre és megjeleníti egy (az illesztőprogram és a bővítőhely attribútumértékeinek megadásával megkapott) PKCS #11 token által támogatott elérhető PKCS #11 objektumokat.

Kötelező attribútumok

```
action_name = "LIST_OBJECTS"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
```

Választható attribútumok

```
p11_login = "<logikai érték>"
p11_label = "<karakterlánc>"
p11_class = "<PKCS#11 objektumosztály>"
p11_private = "<logikai érték>"
p11_trusted = "<logikai érték>"
p11_sensitive = "<logikai érték>"
start_gui = "<logikai érték>"
```

Ahol <*PKCS#11 objektumosztály*> értéke a következők egyike az RSA-ban megadott PKCS #11 specifikáció szerint:

```
CKO_DATA
CKO_CERTIFICATE
CKO_PUBLIC_KEY
CKO_PRIVATE_KEY
CKO_SECRET_KEY
CKO_HW_FEATURE
CKO_DOMAIN_PARAMETERS
CKO_MECHANISM
CKO_VENDOR_DEFINED
```

Példa

```
[p11km_cmd_list_private_objs]
action_name = "LIST_OBJECTS"
p11_login = "TRUE"
p11_private = "TRUE"
p11_driver = "AIX"
p11_slot = "5"
```

PKCS #11 token felhasználó PIN módosítása

Módosítja egy PKCS #11 token felhasználónak a tokenbe bejelentkezéshez használt PIN kódját.

Kötelező attribútumok

```
action_name = "CHANGE_USER_PIN"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
```

Választható attribútumok

```
start_gui = "<logikai érték>"
```

Példa

```
[p11km_cmd_change_my_pin]
action_name = "CHANGE_USER_PIN"
p11_slot = "1337"
p11_driver = "IBM_4764"
```

PKCS #11 objektumok törlése

Törli a PKCS #11 objektumokat. Az objektumok törlésére a **LIST_OBJECTS** parancs futtatásával eredményként kapott objektumok számozott listája alapján és a következő attribútumokkal rendelkező sablon használatával kerül sor:

```
p11_label = "<karakterlánc>"
p11_class = "<PKCS#11 objektumosztály>"
p11_private = "<logikai érték>"
p11_trusted = "<logikai érték>"
p11_sensitive = "<logikai érték>"
p11_login = "<logikai érték>"
```

FIGYELEM: Mivel a token állapota és a konzisztencia nem marad meg a kötegelt feldolgozások között, véletlenül sor kerülhet az objektumok törlésére. Az objektumok felsorolási sorrendje módosul, ha egy objektum felsorolása és törlése közötti időben ugyanazon a tokenen futó más folyamatok objektumokat vesznek fel vagy törölnek.

Kötelező attribútumok

```
action_name = "DELETE_OBJECTS"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
p11_objects = "<CSV>"
```

Ahol <*CSV*> értéke vagy ALL (minden token objektum) vagy a megjelenés sorrendjében számozott objektumoknak megfelelő pozitív egész érték a következő választható attribútumokkal.

Választható attribútumok

```
p11_label = "<karakterlánc>"
p11_class = "<PKCS#11 objektumosztály>"
p11_private = "<logikai érték>"
p11_trusted = "<logikai érték>"
p11_sensitive = "<logikai érték>"
p11_login = "<logikai érték>"
start_gui = "<logikai érték>"
```

Példa

```
[p11km_cmd_delete_seven_objects]
action_name = "DELETE_OBJECTS"
p11_slot = "0"
p11_driver = "AIX"
p11_objects = "1,5,10,11,12,27,33"
p11_login = "TRUE"
```

PKCS #11 objektumok áthelyezése:

PKCS #11 objektumok áthelyezése. Az objektumok áthelyezésére a **LIST_OBJECTS** parancs futtatásával eredményként kapott objektumok számozott listája alapján és ugyanannak a sablonnak a használatával kerül sor.

FIGYELEM: Mivel a token állapota és a konzisztencia nem marad meg a kötegelt feldolgozások között, véletlenül sor kerülhet az objektumok áthelyezésére. Az objektumok felsorolási sorrendje módosul, ha egy objektum felsorolása és törlése közötti időben ugyanazon a tokenen futó más folyamatok objektumokat vesznek fel vagy helyeznek át.

Kötelező attribútumok

```
action_name = "MOVE_OBJECTS"
##########################################
###### Source Token Identification: ######
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
##########################################
###### Target Token Identification: ######
p11_driver_target = "<illesztőprogram neve>"
p11_slot_target = "<bővítőhely száma>"
##########################################
###### Objects being moved to target: ######
p11_objects = "<CSV>"
```

Választható attribútumok

```
p11_label = "<karakterlánc>"
p11_class = "<PKCS#11 objektumosztály>"
p11_private = "<logikai érték>"
p11_trusted = "<logikai érték>"
p11_sensitive = "<logikai érték>"
p11_login = "<logikai érték>"
start_gui = "<logikai érték>"
```

Példa

```
[p11km_cmd_move_three_objects]
action_name = "MOVE_OBJECTS"
p11_slot = "0"
p11_slot_target = "1"
p11_driver = "AIX"
p11_driver_target = "AIX"
p11_objects = "15,20,60"
p11_login = "FALSE"
```

PKCS #11 objektumok másolása

Átmásolja a PKCS #11 objektumokat. Az objektumok átmásolására a **LIST_OBJECTS** parancs futtatásával eredményként kapott objektumok számozott listája alapján és ugyanannak a sablonnak a használatával kerül sor.

FIGYELEM: Mivel a token állapota és a konzisztencia nem marad meg a kötegelt feldolgozások között, véletlenül sor kerülhet az objektumok másolására. Az objektumok felsorolási sorrendje módosul, ha egy objektum felsorolása és törlése közötti időben ugyanazon a tokenen futó más folyamatok objektumokat vesznek fel vagy másolnak át.

Kötelező attribútumok

```
action_name = "COPY_OBJECTS"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
p11_driver_target = "<illesztőprogram neve>"
p11_slot_target = "<bővítőhely száma>"
p11_objects = "<CSV>"
```

Választható attribútumok

```
p11_label = "<karakterlánc>"
p11_class = "<PKCS#11 objektumosztály>"
p11_private = "<logikai érték>"
p11_trusted = "<logikai érték>"
p11_sensitive = "<logikai érték>"
p11_login = "<logikai érték>"
start_gui = "<logikai érték>"
```

Példa

```
[p11km_cmd_copy_one_private_object]
action_name = "COPY_OBJECTS"
p11_slot = "0"
p11_slot_target = "1"
p11_driver = "AIX"
p11_driver_target = "AIX"
p11_objects = "3"
p11_login = "TRUE" ## A PRIVÁT OBJEKTUMOK KEZELÉSÉHEZ KELL.
```

PKCS #11 objektumok exportálása és mentése fájlba

Exportálja és elmenti a PKCS #11 objektumokat. Az objektumok exportálására és biztonsági mentésére a **LIST_OBJECTS** parancs futtatásával eredményként kapott objektumok számozott listája alapján és ugyanannak a sablonnak a használatával kerül sor.

FIGYELEM: Mivel a token állapota és a konzisztencia nem marad meg a kötegelt feldolgozások között, véletlenül sor kerülhet az objektumok exportálására. Az objektumok felsorolási sorrendje módosul, ha egy objektum felsorolása és törlése közötti időben ugyanazon a tokenen futó más folyamatok objektumokat vesznek fel vagy exportálnak.

Kötelező attribútumok

```
action_name = "EXPORT_OBJECTS"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
p11_object_file = "<fájlnév>"
p11_objects = "<CSV>"
```

Választható attribútumok

```
p11_label = "<karakterlánc>"
p11_class = "<PKCS#11 objektumosztály>"
p11_private = "<logikai érték>"
p11_trusted = "<logikai érték>"
p11_sensitive = "<logikai érték>"
p11_login = "<logikai érték>"
start_gui = "<logikai érték>"
```

Példa

```
[p11km_cmd_backup_objects]
action_name = "EXPORT_OBJECTS"
p11_slot = "0"
p11_driver = "AIX"
p11_objects = "ALL"
p11_login = "TRUE"
p11_object_file = "/home/user1/p11km.backup"
```

PKCS #11 objektumok importálása fájlból

Importálja a PKCS #11 exportfájlból létrehozott PKCS #11 objektumokat.

Kötelező attribútumok

```
action_name = "IMPORT_OBJECTS"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
p11_object_file = "<fájlnév>"
```

Választható attribútumok

```
p11_login = "<boolean>" # A PRIVÁT OBJEKTUMOK IMPORTÁLÁSÁHOZ KELL.
start_gui = "<logikai érték>"
```

Példa

```
[p11km_cmd_import_my_backed_up_objects]
action_name = "IMPORT_OBJECTS"
p11_slot = "0"
p11_driver = "AIX"
p11_login = "TRUE"
p11_object_file = "/home/user1/p11km.backup"
```

Saját aláírású igazolás létrehozása

Saját aláírású X.509 igazolást és társított PKCS #11 objektumokat hoz létre egy PKCS #11 tokenen.

Kötelező attribútumok

```
action_name = "CREATE_SSC"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
p11_login = "TRUE"
p11_ssc_label = "<karakterlánc>"
p11_ssc_config = "<openssl konfigurációs fájl>"
```

Ahol az <*openssl konfigurációs fájl*> annak az OpenSSL konfigurációs fájl teljes UNIX elérési útja és fájlneve, amely a saját aláírású igazolás létrehozásában felhasznált értékekkel került feltöltésre.

Választható attribútumok

```
start_gui = "<logikai érték>"
```

Példa

```
[p11km_cmd_self_signed_certificate]
action_name = "CREATE_SSC"
p11_slot = "0"
```

```
p11_driver = "AIX"
p11_login = "TRUE"
p11_ssc_label = "Lab RADIUS Server"
p11_ssc_config = "/etc/radius/EAP-TLS/openssl.cnf"
```

PKCS #10 igazolás aláírási kérés létrehozása

Létrehoz egy PKCS #10 igazolási kérést vagy igazolás aláírási kérést (CSR).

Kötelező attribútumok

```
action_name = "CREATE_CSR"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
p11_login = "TRUE"
p11_csr_label = "<karakterlánc>"
p11_csr_file = "<CSR kimeneti fájl útvonala>"
p11_csr_type = "<DER vagy Base64>'
p11_csr_config = "<openssl konfigurációs fájl>"
```

Ahol <*DER vagy Base64*> egy ASN.1 (DER) kódolású CSR kimeneti fájlt vagy egy Base64 kódolású CSR kimeneti fájlt állít elő, a <*CSR kimeneti fájl útvonala*> pedig a CSR kimenet teljesUNIX útvonalát és fájlnevét azonosítja.

Választható attribútumok

```
start_gui = "<logikai érték>"
```

Példa

```
[p11km_cmd_my_pkcs10_base64]
action_name = "CREATE_SSC"
p11_slot = "0"
p11_driver = "AIX"
p11_login = "TRUE"
p11_csr_label = "Lab RADIUS Server"
p11_csr_type = "Base64"
p11_csr_file = "/etc/radius/EAP-TLS/certreq.b64"
p11_csr_config = "/etc/radius/EAP-TLS/openssl.cnf"
```

A következő kötegelt parancsok érhetők el a PKCS #11 adminisztrációs eszközben (p11admin).

Megjegyzés: A kötegelt parancsok használatához tegye a következőket:

1. Hozzon létre és szerkesszen egy parancsfájlt a következő fejezetben leírtak szerint: “Kötegelt feldolgozás” oldalszám: 181.
2. Hozzon létre új p11km_cmd szakaszokat, amelyek tartalmazzák a használni kívánt kötegelt parancsok attribútumait.

Elérhető PKCS #11 tokenek felsorolása

Jelentést hoz létre és megjeleníti az elérhető PKCS #11 tokenek token és bővítőhely információit.

Kötelező attribútumok

```
action_name = "ADM_LIST_TOKENS"
```

Választható attribútumok

```
start_gui = "<logikai érték>"
```

Ahol <*logikai érték*> értéke vagy TRUE, vagy FALSE

Példa

```
[p11admin_cmd_list_tokens]
action_name = "ADM_LIST_TOKENS"
```

Elérhető PKCS#11 mechanizmusok felsorolása

Jelentést hoz létre és megjeleníti egy (az illesztőprogram és a bővítőhely attribútumértékeinek megadásával megkapott) PKCS #11 token által támogatott elérhető PKCS #11 mechanizmusokat.

Kötelező attribútumok

```
action_name = "ADM_LIST_MECHANISMS"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
```

Ahol <*bővítőhely száma*> pozitív egész érték, az <*illesztőprogram neve*> pedig a következő értékek egyike:

Érték	Leírás
AIX	AIX operációs rendszer titkosítási keretrendszere
IBM_4758_4960	IBM 4758/4960 titkosítási hardveradapterek
IBM_4764	IBM 4764 titkosítási hardveradapterek
Other	Ha OTHER értéket ad meg, akkor meg kell adnia a **p11_driver_path** attribútumot is.

Választható attribútumok

```
start_gui = "<logikai érték>"
```

Kiegészítő attribútumok

```
p11_driver_path = "<a PKCS#11 illesztőprogram útvonala>"
```

Ahol <*PKCS#11 illesztőprogram útvonala*> a parancshoz használt PKCS #11 függvénytár teljes UNIX elérési útja és fájlneve. Ez az attribútum csak akkor adható meg, ha a **p11_driver** attribútum értéke OTHER.

Példa

```
[p11admin_cmd_list_4764_slot_0_mechs]
action_name = "ADM_LIST_MECHANISMS"
p11_driver = "IBM_4764"
p11_slot = "0"
start_gui = "TRUE"
```

PKCS #11 token információinak megjelenítése

Információkat jelenít meg egy PKCS #11 tokenről és a bővítőhelyéről.

Kötelező attribútumok

```
action_name = "ADM_SHOW_TOKEN_INFO"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
```

Választható attribútumok

```
start_gui = "<logikai érték>"
```

Példa

```
[p11admin_cmd]
action_name = "ADM_SHOW_TOKEN_INFO"
p11_slot = "411"
p11_driver = "IBM_4764"
```

PKCS #11 token inicializálása:

Inicializál egy PKCS #11 tokent. Az inicializáció alaphelyzetbe állítja a tokent, töröl minden tárolt PKCS#11 objektumot és adatot, valamint lehetővé teszi a token újracímkézését.

FIGYELEM: Mivel minden PKCS #11 objektum és adat törlésére sor kerül az inicializációs folyamat során, ezért a PKCS #11 token inicializálása előtt győződjön meg arról, hogy már nincs szüksége az objektumokra és adatokra.

Kötelező attribútumok

```
action_name = "ADM_INIT_TOKEN"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>" ## UGYANAZ, MINT 'p11_init_slot'
p11_init_slot = "<bővítőhely száma>" ## UGYANAZ, MINT 'p11_slot'
p11_init_label = "<karakterlánc>" ## ÚJ TOKEN CÍMKE
```

Választható attribútumok

```
start_gui = "<logikai érték>"
```

Példa

```
[p11admin_cmd]
action_name = "ADM_INIT_TOKEN"
p11_slot = "1"
p11_driver = "IBM_4764"
p11_init_slot = "1"
p11_init_label = "ABC Token"
```

PKCS #11 token órájának megtekintése

Megjeleníti egy PKCS #11 token hardveres óráját, ha van olyan.

Kötelező attribútumok

```
action_name = "ADM_CLOCK_VIEW"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
```

Választható attribútumok

```
start_gui = "<logikai érték>"
```

Példa

```
[p11admin_cmd]
action_name = "ADM_CLOCK_VIEW"
p11_slot = "1"
p11_driver = "IBM_4764"
```

PKCS #11 token órájának beállítása

Beállítja egy PKCS #11 token hardveres óráját, ha van olyan.

Kötelező attribútumok

```
action_name = "ADM_CLOCK_SET"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
p11_clock_set = "<óra adatai>"
```

Ahol <*óra adatai*> a jelenlegi UTC dátum és idő a következő formátummal: ÓÓ:PP:MP hh-nn-ÉÉÉÉ

Választható attribútumok

```
start_gui = "<logikai érték>"
```

Példa

```
[p11admin_cmd]
action_name = "ADM_CLOCK_SET"
p11_slot = "1"
p11_driver = "IBM_4764"
p11_clock_set = "23:59:59 12-31-1999"
```

Alaphelyzetbe állítja egy PKCS #11 token felhasználó PIN kódját

Alaphelyzetbe állítja egy PKCS #11 token felhasználó PIN kódját.

Kötelező attribútumok

```
action_name = "ADM_RESET_USER_PIN"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
```

Választható attribútumok

```
start_gui = "<logikai érték>"
```

Példa

```
[p11admin_cmd_change_so_pin]
action_name = "ADM_RESET_USER_PIN"
p11_driver = "AIX"
p11_slot = "0"
```

PKCS #11 token adatvédelmi megbízottjához tartozó PIN kód módosítása

Megváltoztatja egy PKCS #11 token adatvédelmi megbízottjának PIN kódját. Ez a PIN a token adminisztrátori műveleteinek végrehajtásakor használatos.

Kötelező attribútumok

```
action_name = "ADM_CHANGE_SO_PIN"
p11_driver = "<illesztőprogram neve>"
p11_slot = "<bővítőhely száma>"
```

Választható attribútumok

```
start_gui = "<logikai érték>"
```

Példa

```
[p11admin_cmd_change_so_pin]
action_name = "ADM_CHANGE_SO_PIN"
p11_slot = "888"
p11_driver = "IBM_4764"
```

Cserélhető hitelesítési modulok

A cserélhető hitelesítési modul (PAM) keretrendszer lehetővé teszi a rendszeradminisztrátorok számára, hogy többféle hitelesítési mechanizmust is beépítsenek a meglévő rendszerbe cserélhető modulok segítségével.

A PAM használatára felkészített alkalmazásokban *kicserélhetők* a régi technológiák újakra a meglévő alkalmazások módosítása nélkül. E rugalmasság révén a rendszergazdák:

- A rendszer tetszés szerinti hitelesítési szolgáltatását kiválaszthatják egy adott alkalmazáshoz
- Többféle hitelesítési mechanizmust is használhatnak egy adott szolgáltatáshoz
- Új hitelesítési szolgáltatásmodulokat vehetnek fel a meglévő alkalmazások módosítása nélkül.
- Egy korábban beírt jelszót több modul hitelesítéséhez is felhasználhatnak.

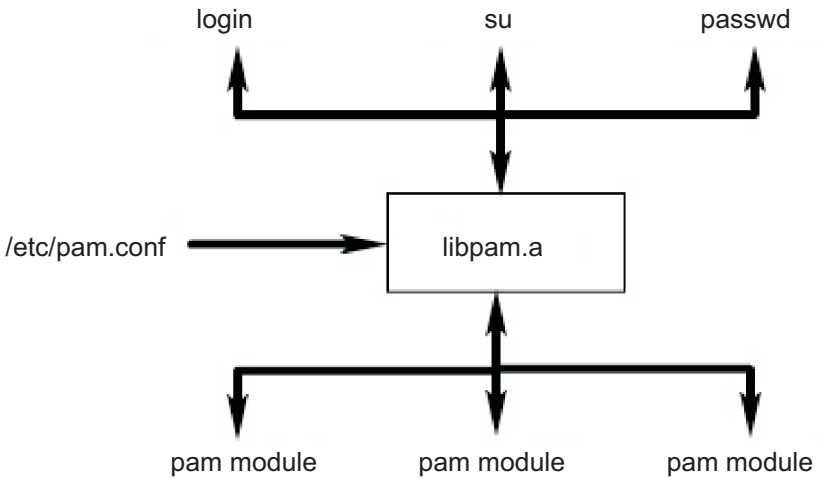
A PAM keretrendszer egy könyvtárból, cserélhető modulokból és egy konfigurációs fájlból áll. A PAM könyvtár valósítja meg a PAM alkalmazásprogram illesztőt (API-t), végzi a PAM tranzakciók kezelését, és hívja meg a cserélhető modulokban definiált PAM szolgáltatásprogram illesztőt (SPI-t). A cserélhető modulok a meghívó szolgáltatás és a konfigurációs fájl bejegyzései alapján dinamikusan töltődnek be a könyvtárba. A sikert nemcsak a cserélhető modul határozza meg, hanem az adott szolgáltatáshoz definiált viselkedés is. Az *egymásra építés* fogalmát használva egy szolgáltatás beállítható több hitelesítési módszerrel végzett hitelesítésre is. Amennyiben támogatják, a modulok beállíthatók egy korábban megadott jelszó használatára, nem pedig újra bekérésére.

A rendszeradminisztrátor az AIX rendszert PAM használatára az /etc/security/login.cfg fájl usw szakaszában található **auth_type** attribútum módosításával konfigurálhatja. Az auth_type = PAM_AUTH beállítás a PAM-ra felkészített parancsokat a hitelesítéshez a PAM API közvetlen meghívására állítja be, nem pedig a korábbi AIX hitelesítési rutinok meghívására. Ezt a konfigurációt futás közben lehet beállítani, az életbe lépéshez nem kell újraindítani a rendszert. Az **auth_type** attribútummal kapcsolatos további információkat a /etc/security/login.cfg fájlleírás tartalmaz. Az alábbi natív AIX parancsok és alkalmazások módosításra kerültek az**auth_type** attribútum felismeréséhez, és fel lettek készítve a PAM hitelesítésre:

- **login**
- **passwd**
- **su**
- **ftp**
- **telnet**
- **rlogin**
- **rexec**
- **rsh**
- **snappd**
- **imapd**

- **dtaction**
- **dtlogin**
- **dtsession**

Az alábbi ábra a PAM használatára képes alkalmazások, a PAM könyvtár, a konfigurációs fájl és a PAM használatára beállított rendszeren található PAM modulok közötti együttműködést mutatja be. A PAM támogatással rendelkező alkalmazások a PAM könyvtárban található PAM API-t hívják. A könyvtár a konfigurációs fájl bejegyzése alapján meghatározza, melyik modult kell betölteni, majd meghívja a modul PAM SPI-jét. A kommunikáció a PAM modul és az alkalmazás között az alkalmazásba beépített párbeszéd funkcióval történik. A modul sikere vagy sikertelensége, valamint a konfigurációs fájlban megadott viselkedés együttese határozza meg, hogy további modulokat be kell-e tölteni. Ha igen, a folyamat folytatódik; ha nem, az eredmény visszakerül az alkalmazáshoz.



3. ábra: PAM keretrendszer és entitások. Ez az ábra azt mutatja be, hogy a PAM támogatással rendelkező alkalmazás hogyan használja a PAM könyvtárat a megfelelő PAM modul elérésére.

PAM könyvtár

Az /usr/lib/libpam.a PAM könyvtár tartalmazza azt a PAM API-t, amely közös illesztőként szolgál minden PAM alkalmazás számára, illetve amely a modulok betöltését vezérli.

A modulok betöltését a PAM könyvtár vezérli az /etc/pam.conf fájlban meghatározott egymásra építési szabályok szerint.

Az alábbi PAM API funkciók hívják meg a PAM modul megfelelő PAM SPI-jét. A pam_authenticate API például a pam_sm_authenticate SPI-t hívja meg egy PAM modulban.

- pam_authenticate
- pam_setcred
- pam_acct_mgmt
- pam_open_session
- pam_close_session
- pam_chauthtok

A PAM könyvtárban ezenfelül számos keret API is található, amelyekkel az alkalmazások meghívhatják a PAM modulokat, illetve információkat adhatnak át nekik. Az alábbi táblázat az AIX rendszerben megvalósított PAM keretrendszer API-kat és azok funkcióit mutatja be:

PAM keretrendszer API	Funkció
pam_start	PAM szekció létrehozása
pam_end	PAM szekció lezárása
pam_get_data	Modulspecifikus adatok lekérdezése
pam_set_data	Modulspecifikus adatok beállítása
pam_getenv	Definiált PAM környezeti változó értékének lekérdezése
pam_getenvlist	PAM környezeti változók és azok értékeinek lekérdezése
pam_putenv	PAM környezeti változó beállítása
pam_get_item	Általános PAM információk lekérdezése
pam_set_item	Általános PAM információk beállítása
pam_get_user	Felhasználónév lekérdezése
pam_strerror	PAM szabvány hibaüzenet lekérése

PAM modulok

A PAM modulok lehetővé teszik egy rendszeren többféle hitelesítési mechanizmus használatát.

Egy adott PAM modulnak négy modultípus közül legalább egyet meg kell valósítania. Az alábbiakban bemutatjuk a modultípusokat, valamint a modultípus-megfeleléshez szükséges, hozzájuk tartozó PAM SPI-ket.

Hitelesítési modulok

Hitelesítik a felhasználókat, valamint beállítják, frissítik, vagy megsemmisítik a hitelesítési adatokat. Ezek a modulok a felhasználót a hitelesítésük és a hitelesítési adataik alapján azonosítják.

Hitelesítési modul funkciók:

- pam_sm_authenticate
- pam_sm_setcred

Fiókkezelési modulok

A hitelesítési modul azonosítása után megállapítják a felhasználói fiók és a hozzáférés érvényességét. Az ilyen típusú modulok által végzett ellenőrzések közé tartozik jellemzően a fióklejárat, illetve a jelszókorlátozások ellenőrzése.

A fiókkezelési modul által megvalósított funkció:

- pam_sm_acct_mgmt

Szekciókezelési modulok

Felhasználói szekciók kezdeményezése és lezárása. Ezenfelül esetleg támogatják a szekció megfigyelését is.

A szekciókezelési modulok az alábbi funkciókat kínálják:

- pam_sm_open_session
- pam_sm_close_session

Jelszókezelési modulok

Jelszómódosítást és a vonatkozó attribútumok karbantartását végzik.

A jelszókezelési modulok az alábbi funkciókat kínálják:

- pam_sm_chauthtok

PAM konfigurációs fájl

Az /etc/pam.conf konfigurációs fájl szolgáltatásbejegyzéseket tartalmaz minden egyes PAM modultípushoz. Célja, hogy a szolgáltatásokat egy meghatározott modul-útvonalon keresztül irányítsa.

A fájl bejegyzései az alábbi, fehér szóközökkel határolt mezőkből állnak:

szolgáltatásnév modultípus vezérlőparaméter modul_elérési_út modul_opciók

A mezők leírása:

szolgáltatásnév
A szolgáltatás nevét adja meg. Az OTHER kulcsszó a bejegyzésben nem megadott alkalmazások által használandó alapértelmezett modult határozza meg.

modultípus
A szolgáltatás modultípusát adja meg. Az érvényes modultípusok: **auth**, **account**, **session** vagy **password**. Egy adott modul egy vagy több modul típushoz biztosít támogatást.

vezérlőparaméter
A szolgáltatás egymásra építési viselkedését határozza meg. A használható vezérlőparaméterek: required (kötelező), requisite (szükséges), sufficient (elégséges) vagy optional (elhagyható).

modul_elérési_út

Megadja a szolgáltatáshoz betöltendő modult. A *modul_elérési_út* érvényes értékei megadhatók a modul teljes elérési útjaként vagy csak a modul nevével. Ha a modul teljes elérési útja van van megadva, akkor a PAM könyvtár azt a *modul_elérési_út* értéket használja a 32 bites szolgáltatások betöltéséhez, illetve a 64 alkönyvtárat használja a 64 bites szolgáltatásokhoz. Ha a modul teljes elérési útja nincs megadva, akkor a PAM könyvtár hozzáfűzi a /usr/lib/security előtagot (32 bites szolgáltatások esetén) vagy a /usr/lib/security/64 (64 bites szolgáltatások esetén) a modulnévhez.

modul_opciók
A szolgáltatásmoduloknak átadható opciók szóközökkel elválasztott listája. A mező értéke függ a *modul_elérési_út* mezőben megadott modul opcióitól. A mező kitöltése nem kötelező.

A rosszul kialakított bejegyzéseket, illetve a **modultípus** és **vezérlőparaméter** mezők helytelen értékeit a PAM könyvtár figyelmen kívül hagyja. A megjegyzésekre utaló kettőskereszt (#) karakterrel kezdődő sorok szintén figyelmen kívül hagyódnak.

A PAM támogatja a "egymásra építésnek" nevezett alapelvet, ami lehetővé teszi, hogy több mechanizmust is használjon a szolgáltatásokhoz. Az egymásra építés a konfigurációs fájlban úgy valósítható meg, ha több bejegyzés készül ugyanazzal a **modultípus** mezővel egy szolgáltatáshoz. A modulok a szolgáltatáshoz a fájlba beírt sorrendben kerülnek meghívásra, a végeredményt pedig az egyes bejegyzésekhez megadott **vezérlőparaméter** mező fogja meghatározni. A **vezérlőparaméter** mező érvényes értékei és a modulcsomag megfelelő viselkedése az alábbiak lehetnek:

A control_flag mező értéke	**Viselkedés**
required	A csomag minden kötelező modulja "sikeres" eredményt kell, hogy adjon. Ha a kötelező modulok bármelyike sikertelen, akkor továbbra is meghívásra kerül az összes kötelező modul, de az első sikertelen kötelező modul eredménye kerül visszaadásra.
requisite	Hasonlít a required modulhoz azzal a kivétellel, hogy ha egy requisite modul hibába ütközik, akkor a verem többi modulja nem kerül feldolgozásra, és azonnal visszaadja az első hibakódot a required vagy requisite modulból.
sufficient	Ha egy elégségesnek jelölt modul sikeres, és korábban egyetlen kötelező vagy elégséges modul sem volt sikertelen, akkor a további modulok figyelmen kívül hagyódnak, és a "sikeres" érték kerül visszaadásra.
optional	Ha a csomag egyetlen modulja sem kötelező, és egyetlen elégséges modul sem volt sikeres, akkor legalább egy elhagyható modulnak sikeresnek kell lennie. Ha a csomag bármelyik másik modulja sikeres, akkor az elhagyható modul sikertelensége figyelmen kívül hagyódik.

Az alábbi /etc/pam.conf alkészlet egy példa a egymásra építésre az auth modul típusban a login szolgáltatáshoz.

```
#
# PAM konfigurációs fájl  /etc/pam.conf
#

# Hitelesítés-kezelés
login   auth     required       /usr/lib/security/pam_ckfile    file=/etc/nologin
login   auth     required       /usr/lib/security/pam_aix
login   auth     optional       /usr/lib/security/pam_test      use_first_pass
OTHER   auth     required       /usr/lib/security/pam_prohibit
```

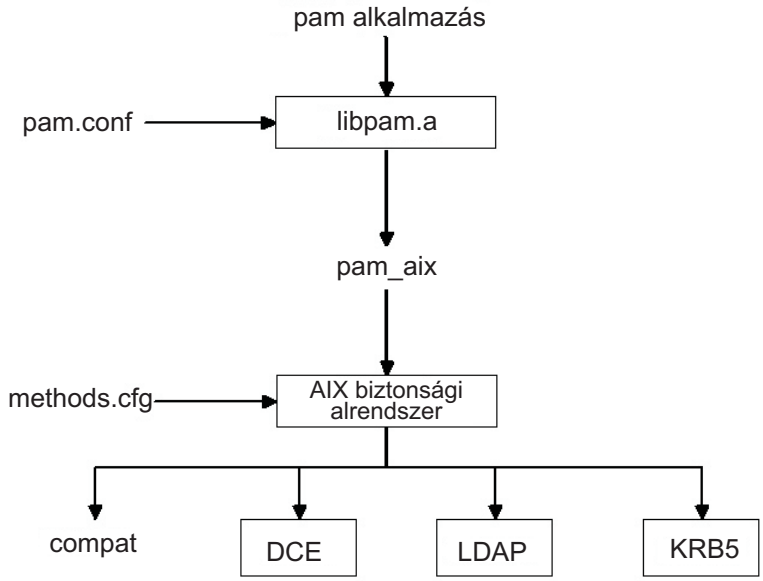
A minta konfigurációs fájl három bejegyzést tartalmaz a bejelentkezési szolgáltatáshoz. Ha a pam_ckfile és pam_aix is required paraméterként van megadva, akkor mindkét modul futtatásra kerül, és a sikerességhez mindkét modul végrehajtásának sikeresnek kell lennie. A pam_test modul harmadik bejegyzése elhagyható, sikere vagy sikertelensége nem befolyásolja azt, hogy a felhasználó képes-e bejelentkezni. A pam_test modul use_first_pass opciója megköveteli a korábban már beírt jelszó használatát az új bekérése helyett.

Az OTHER kulcsszó szolgáltatásnévként megadásával alapértelmezett érték állítható be minden más, a konfigurációs fájlban nem kifejezetten megadott szolgáltatáshoz. Az alapértelmezett érték beállítása garantálja, hogy egy adott modultípust mindig legalább egy modul lefed. Ebben a példában a login szolgáltatáson kívül minden szolgáltatás hibába fog ütközni, mivel a pam_prohibit modul PAM hibát ad vissza minden hívásnál.

pam_aix modul

A pam_aix module egy olyan PAM modul, amely PAM-ra felkészített alkalmazások számára hozzáférést biztosít az AIX biztonsági szolgáltatásaihoz olyan felületek segítségével, amelyek az egyenértékű AIX szolgáltatásokat hívják meg, amennyiben ezek léteznek.

Ezeket a szolgáltatásokat azután egy betölthető hitelesítési modul vagy az AIX beépített funkció hajtja végre a felhasználó-definíciók és a methods.cfg fájl megfelelő beállításai alapján. Az AIX szolgáltatás végrehajtása közben kapott minden hibakód a megfelelő PAM hibakóddá fordítódik le.



4. ábra: PAM alkalmazás és az AIX biztonsági alrendszere közötti út

Az alábbi ábra azt az utat mutatja, amelyet egy PAM alkalmazás API hívása jár be, ha az /etc/pam.conf fájlban be van állítva a pam_aix modul használata. Az ábrán is látható módon, az integráció révén a felhasználók hitelesíthetők bármelyik betölthető hitelesítési modullal (DCE, LDAP vagy KRB5) vagy AIX fájlokkal (compat).

A pam_aix modul az /usr/lib/security könyvtárba kerül telepítésre. A pam_aix modul integrációjához az /etc/pam.conf fájlt be kell állítani a modul használatára. Bár a modulok egymásra építése továbbra is rendelkezésre áll, az alábbi /etc/pam.conf fájlban nem látható:

```
#
# Hitelesítés-kezelés
#
OTHER   auth     required       /usr/lib/security/pam_aix

#
# Hitelesítés-kezelés
#
```

```
OTHER   account  required       /usr/lib/security/pam_aix

#
# Hitelesítés-kezelés
#
OTHER   session  required       /usr/lib/security/pam_aix

#
# Hitelesítés-kezelés
#
OTHER   password required       /usr/lib/security/pam_aix
```

A pam_aix modul pam_sm_authenticate, pam_sm_chauthok és pam_sm_acct_mgmt SPI funkciókhoz rendelkezik megvalósításokkal. A pam_sm_setcred, pam_sm_open_session és pam_sm_close_session SPI-k is meg vannak valósítva a pam_aix modulban, de ezek az SPI függvények csak PAM_SUCCESS eredményt adnak vissza.

Az alábbiakban vázlatosan feltüntetjük a PAM SPI hívások leképezését az AIX biztonsági alrendszerére:

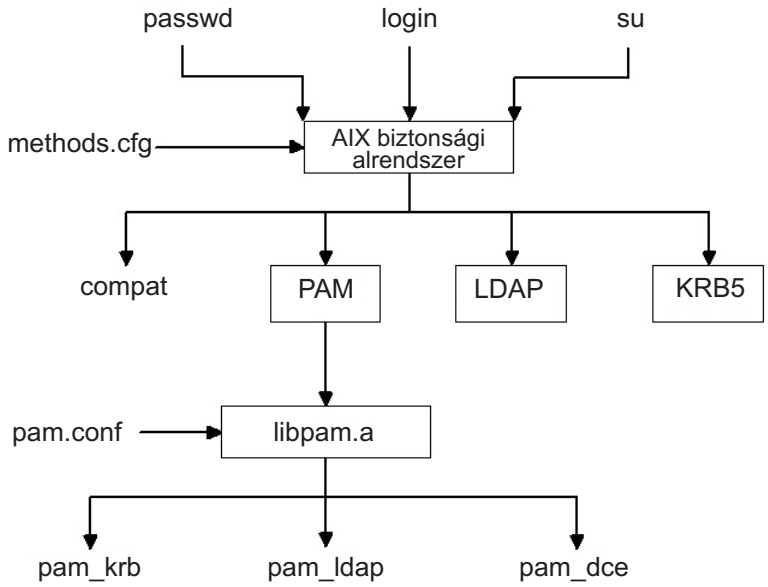
```
 PAM SPI                    AIX
=========                   =====
pam_sm_authenticate   -->  authenticate
pam_sm_chauthtok      -->  passwdexpired, chpass
                           Megjegyzés: a passwdexpired csak akkor kerül
                           ellenőrzésre, ha a PAM_CHANGE_EXPIRED_AUTHTOK
                           jelző át van adva.
pam_sm_acct_mgmt      -->  loginrestrictions, passwdexpired
pam_sm_setcred        -->  Nincs megfelelő leképezés, PAM_SUCCESS érték kerül visszaadásra
pam_sm_open_session   -->  Nincs megfelelő leképezés, PAM_SUCCESS érték kerül visszaadásra
pam_sm_close_session  -->  Nincs megfelelő leképezés, PAM_SUCCESS érték kerül visszaadásra
```

Az AIX biztonsági alrendszernek átadni kívánt adatok átadhatók a pam_set_item funkcióval a modul használata előtt, vagy ha még nem léteznek az adatok, akkor a pam_aix modul bekéri az adatokat.

PAM betölthető hitelesítési modul

AIX biztonsági szolgáltatások beállíthatók úgy, hogy a PAM modulokat meglévő AIX betölthető hitelesítési modul keretrendszer segítségével hívja meg.

Ha a /usr/lib/security/methods.cfg fájl megfelelően van beállítva, akkor a PAM betölthető modul az AIX biztonsági szolgáltatásait (passwd, login és így tovább) a PAM könyvtár felé továbbítja. A PAM könyvtár ellenőrzi az /etc/pam.conf fájlt és megállapítja, melyik PAM modult kell használnia, majd elvégzi a megfelelő PAM SPI-hívást. A PAM visszatérési értékei az AIX hibakódjaira képeződnek le, majd visszaadódnak a hívó programnak.



5. ábra: Az AIX biztonsági szolgáltatásai és a PAM modul közötti út

Az alábbi ábra azt mutatja be, milyen utat jár be egy AIX biztonsági szolgáltatás hívás a PAM helyes beállítása esetén. A bemutatott PAM modulok (pam_krb, pam_ldap és pam_dce) külső megoldások példáiként kerülnek megjelenítésre.

A csak hitelesítést végző PAM modul az /usr/lib/security könyvtárban kerül telepítésre. A PAM modult egy adatbázissal egyesítve alakítható ki egy összetett betöltésű modul. Az alábbi példa bemutatja, milyen szakaszokat kell beírni a methods.cfg fájlba egy összetett PAM modul készítéséhez a files nevű adatbázissal. A **db** attribútum BUILTIN kulcsszava az adatbázist UNIX fájlokként adja meg.

```
PAM:
        program = /usr/lib/security/PAM

PAMfiles:
        options = auth=PAM,db=BUILTIN
```

A felhasználók létrehozása és módosítása ezek után az adminisztrációs parancsok -R kapcsolójával végezhető el, illetve a felhasználó létrehozása után a *SYSTEM* attribútum beállításával. Például:

```
mkuser -R PAMfiles SYSTEM=PAMfiles registry=PAMfiles pamuser
```

Ennek hatására az AIX biztonsági szolgáltatásai (login, passwd és hasonlók) felé irányuló hívások a PAM betöltő modulját használják hitelesítésre. Bár a példában a **files** adatbázist használtuk, ha telepítve vannak, más adatbázisok, például az LDAP is használható. A felhasználók a fentiekben ismertetett módon létrehozásával az AIX biztonsági rendszerei az alábbi módon képződnek le PAM API hívásokra:

```
 AIX                    PAM API
=====                   =========
authenticate       -->  pam_authenticate
chpass             -->  pam_chauthtok
passwdexpired      -->  pam_acct_mgmt
passwdrestrictions -->  Nincs megfelelő leképezés, "sikeres" érték kerül visszaadásra
```

Az /etc/pam.conf fájl testreszabásával a PAM API hívásai a kívánt PAM modulhoz irányíthatók hitelesítésre. A hitelesítési mechanizmus további finomításához verem valósítható meg.

Az AIX egy biztonsági szolgáltatása által bekért adatok továbbításra kerülnek a PAM felé a pam_set_item funkcióval, mivel a PAM-ból magából nem lehet felhasználói párbeszédablakokat megnyitni. A PAM modullal integrált használatra készült PAM modulok az összes adatot pam_get_item hívásokkal kell, hogy begyűjtsék, és nem szabad, hogy adatokat kérjenek be a felhasználótól, ez ugyanis a biztonsági szolgáltatás dolga.

Hurokfelismerési szolgáltatás is rendelkezésre áll az olyan esetleges hibák azonosításához, amikor egy AIX biztonsági szolgáltatás a PAM-hoz van irányítva, majd a PAM modul az AIX biztonsági szolgáltatását próbálja meghívni a művelet elvégzéséhez. Egy ilyen hurok észlelése a művelet azonnali sikertelenségét eredményezi.

Megjegyzés: A /etc/pam.conf fájl *nem* írható be úgy, hogy az AIX biztonsági szolgáltatás PAM integrációja során használja a pam_aix modult egy PAM modulhoz, mivel ez hurkot eredményez.

PAM modul felvétele

Több hitelesítési mechanizmus engedélyezéséhez hozzáadhat egy PAM modult.

1. Helyezze a modul 32 bites változatát a /usr/lib/security könyvtárba, a 64 bites változatot pedig a /usr/lib/security/64 könyvtárba.
2. Állítsa be a tulajdonjogot root-ra, a jogosultságokat pedig 555-re. A PAM könyvtár nem tölti be a root felhasználó tulajdonában lévő modulokat.
3. Frissítse az /etc/pam.conf konfigurációs fájlt, hogy az a modult tartalmazza a kívánt szolgáltatásnevekben.
4. Próbálja ki az érintett szolgáltatásokat, hogy helyesen működnek-e. Ne jelentkezzen ki a rendszerből addig, amíg nem hajtott végre egy bejelentkezési tesztet.

A /etc/pam.conf fájl módosítása

A /etc/pam.conf fájl módosítása előtt néhány dolgot át kell gondolni.

A /etc/pam.conf konfigurációs fájl módosításakor vegye figyelembe a következő követelményeket:

- A fájl tulajdonosának mindig a root felhasználónak a security csoportnak kell lennie. A fájl jogosultságának 644-nek kell lennie, hogy mindenki olvashassa, de csak a root felhasználó módosíthassa.
- A nagyobb biztonság érdekében fontolja meg az egyes PAM használatára képes szolgáltatások külön beállítását a pam_prohibit modul OTHER szolgáltatás kulcsszóra állításával.
- Olvassa el az egyes modulokhoz mellékelt dokumentációt, és határozza meg, hogy mely vezérlőparaméterek és beállítások használhatók és mi a hatásuk.
- Gondosan válassza meg a modulok és vezérlőparaméterek sorrendjét, figyelembe véve a required (kötelező), requisite (szükséges), sufficient (megfelelő) és optional (elhagyható) vezérlőparaméterek viselkedését az egymásra épített modulokban.

Megjegyzés: A PAM konfigurációs fájl helytelen beállítása olyan rendszert eredményezhet, amelybe nem lehetséges bejelentkezni, mivel a konfiguráció minden felhasználóra vonatkozik, így a root felhasználóra is. A fájl módosításainak elvégzése után mindig vizsgálja meg az érintett alkalmazásokra gyakorolt hatást, még mielőtt kilépne a rendszerből. Az olyan rendszert, amelybe nem lehet bejelentkezni, a rendszer karbantartási módban elindításával, majd az /etc/pam.conf konfigurációs fájl kijavításával lehet helyreállítani.

PAM hibakeresés engedélyezése

A Cserélhető hitelesítési modul (PAM) függvénytár hibakeresési információkat biztosíthat a végrehajtás során. Ha engedélyezte, hogy a rendszer hibakeresési kimenetet gyűjtsön, akkor az így összegyűjtött információk használhatók a PAM-API hívások nyomon követésére, illetve az aktuális PAM beállítás hibahelyeinek meghatározására.

A PAM hibakeresési kimenet engedélyezéséhez tegye a következőket:

1. Hozzon létre egy pam_debug nevű üres fájlt az /etc/pam_debug könyvtárban a **touch** paranccsal, ha a fájl nem létezik. A PAM függvénytár megkeresi az /etc/pam_debug fájlt, és engedélyezi a syslog kimenetet, ha megtalálta azt.
2. Az /etc/syslog.conf fájl szerkesztésével azonosítson egy fájlt, ahová az auth syslog üzeneteket naplózni fogja a kívánt prioritási szinten. Például ahhoz, hogy PAM hibakeresés-szintű üzeneteket küldjön a /var/log/auth.log fájlba, vegye fel a következő új sort a syslog.conf fájlban:

   ```
   *.debug /var/log/auth.log
   ```

3. Hozza létre a(z)2 lépésben azonosított /var/log/auth.log fájlt a **touch** paranccsal, ha az még nem létezik.
4. A syslogd démon újraindításához, hogy a konfigurációs módosításokat felismerje a rendszer, tegye a következőket:
   a. Állítsa le a syslog démont a következő paranccsal:

```
stopsrc -s syslogd
```

b. Indítsa el a syslog démont a következő paranccsal:

```
startsrc -s syslogd
```

A PAM alkalmazásújraindulása után a hibakeresési üzeneteket a rendszer abba a kimeneti fájlba gyűjti, amely az /etc/syslog.conf konfigurációs fájlban meg lett határozva.

OpenSSH és Kerberos v5 támogatás

A Kerberos egy olyan hitelesítési mechanizmus, amely biztonságos hitelesítést nyújt a hálózati felhasználók számára. A kliensek és szerverek között folyó hitelesítési üzenetek titkosításával megakadályozza nyílt szövegű jelszavak átvitelét a hálózaton. Emellett a Kerberos egy tokenek vagy hitelesítési adatok felügyeletére alapuló felhatalmazási rendszert is biztosít.

A felhasználók Kerberos alapú hitelesítéséhez a felhasználó a **kinit** parancs futtatásával kapja meg a kezdeti hitelesítési adatokat egy kulcselosztó központnak (KDC) nevezett központi Kerberos szerverről. A KDC ellenőrzi a felhasználót, és átadja a felhasználónak a kezdeti hitelesítési adatokat, más néven egy jegymegadási jegyet (TGT). A felhasználó ezután úgy indíthat távoli bejelentkezési szekciókat Kerberosra felkészített Telnet vagy OpenSSH felhasználásával, hogy a Kerberos hitelesíti a felhasználót a felhasználó meghatalmazásának lekérésével a KDC-től. A Kerberos ezt a hitelesítést felhasználói interakció nélkül végzi, ily módon a felhasználónak nem kell jelszót beírnia a bejelentkezéshez. A Kerberos IBM-es változatának neve Hálózati hitelesítési szolgáltatás (NAS). A NAS az AIX bővítőcsomag CD lemezekről telepíthető. A krb5.client.rte és krb5.server.rte csomagokban található. Az OpenSSH 3.6 a 2003. júliusi kiadással kezdődően támogatja a Kerberos 5 hitelesítést és a NAS 1.3 felhatalmazást.

Az OpenSSH 3.8 és újabb változatai támogatják a Kerberos 5 hitelesítést és a NAS 1.4 változaton keresztüli hitelesítést. A NAS (Kerberos) előző változatairól való áttérést az OpenSSH frissítése előtt kell elvégezni. Az OpenSSH 3.8.x változata csak a NAS 1.4 vagy ennél újabb változatával működik.

Az OpenSSH AIX rendszerekre készült változata választható módszerként tartalmazza a Kerberos hitelesítést. Ha a Kerberos könyvtárak nincsenek telepítve a rendszerre az OpenSSH futása során, akkor a Kerberos hitelesítés kimarad, és az OpenSSH a következő beállított hitelesítési módszerrel (például AIX hitelesítés) próbálkozik.

A Kerberos telepítése után ajánlott a Kerberos dokumentáció átböngészése, mielőtt a Kerberos szerverek beállításához kezdene. A Kerberos telepítésével és felügyeletével kapcsolatos információkat a /usr/lpp/krb5/doc/html/*lang*/ADMINGD.htm címen található *IBM Network Authentication Service for AIX 1.3: Adminisztrátori és felhasználói kézikönyv* kiadványban talál.

Kapcsolódó tájékoztatás:

OpenSSH

OpenSSH telepítőkészletek

Az OpenSSH telepítőkészletek telepítéséhez tegye a következőket:

1. Navigáljon az AIX Web Download Pack Programs webhelyre.

   Megjegyzés: Az OpenSSH képfájl az AIX alap adathordozó részeként kerül szállításra, azonban a képfájl nincs telepítve alapértelmezés szerint.
2. Kattintson a **Downloads** hivatkozásra az Additional information részen.
3. A rendelkezésre álló csomagok eléréséhez jelentkezzen be az azonosítójával és jelszavával.
4. Válassza ki az **OpenSSH** lehetőséget, és kattintson a **Continue** gombra.
5. A csomag letöltéséhez fogadja el a licencszerződést.
6. Bontsa ki a csomagot az **uncompress** *csomagnév* paranccsal. Például:

   ```
   uncompress OpenSSH_6.0.0.6203.tar.Z
   ```
7. Bontsa ki a tar állományt a **tar -xvf** *csomagnév* paranccsal. Például:

   ```
   tar -xvf OpenSSH_6.0.0.6203.tar
   ```

8. Futtassa az **inutoc** parancsot.
9. Futtassa a **smitty install** parancsot.
10. Válassza a **Szoftver telepítése és frissítése** menüpontot.
11. Válassza a **Telepített szoftver frissítése a legfrissebb szintre (Összes frissítése)** menüpontot.
12. Írjon be egy pontot (.) a **Szoftver BEMENETI eszköze / könyvtára** mezőbe, majd nyomja meg az Entert.
13. Görgesse le a képernyőt az **Új licencszerződések ELFOGADÁSA** szövegig, majd a **Tab** billentyűvel váltsa át a mezőt **Igen**-re.
14. A telepítés megkezdéséhez kétszer nyomja meg az Enter billentyűt.

Az OpenSSH telepítőkészletek alapszintűek, nem ideiglenes programjavítások (PTF). A telepítéskor az előző verzió minden kódját felülírja az új verzió képe.

OpenSSH fordítás beállítása

Az alábbi információk az OpenSSH kód AIX alatti fordítását mutatják be.

Az OpenSSH AIX Version 6.1 rendszeren végzett beállításakor az alábbihoz hasonló kimenet jelenik meg:

```
OpenSSH has been configured with the following options:
                     User binaries: /usr/bin
                   System binaries: /usr/sbin
               Configuration files: /etc/ssh
                   Askpass program: /usr/sbin/ssh-askpass
                      Manual pages: /usr/man
                          PID file: /etc/ssh
  Privilege separation chroot path: /var/empty
            sshd default user PATH: /usr/bin:/bin:/usr/sbin:/sbin:/usr/
                                    local/bin

                    Manpage format: man
                       PAM support: yes
                     OSF SIA support: no
                    KerberosV support: yes
                    Smartcard support: no
                      SELinux support: no
                        S/KEY support: no
              TCP Wrappers support: yes
              MD5 password support: no
                   libedit support: no
  Solaris process contract support: no
           Solaris project support: no
       IP address in $DISPLAY hack: no
            Translate v4 in v6 hack: no
                  BSD Auth support: no
              Random number source: OpenSSL internal ONLY

              Host: powerpc-ibm-aix6.1.0.0
          Compiler: cc
    Compiler flags: -bloadmap:file -qnostdinc -qnolm -qlist -qsource -qattr=full
    Preprocessor flags: -I/gsa/ausgsa/projects/o/openssh/freeware5/openssl-0.9.8r/
                include -I/gsa/ausgsa/projects/o/openssh/zlib -I/usr/include

          Linker flags: -L/gsa/ausgsa/projects/o/openssh/freeware5/
                    lib -L/gsa/ausgsa/projects/o/openssh/zlib -L/usr/include
                    -Wl,-blibpath:/usr/lib:/lib
             Libraries: -lcrypto -lz -lc  -lcrypt -lefs   -lwrap -lpam -ldl
```

Megjegyzés: Az AIX Version 6.1 és az AIX Version 7.1 fordítási beállításai hasonlóak, mivel mindkét változat bináris fájljai azonosak.

OpenSSH használata Kerberosszal

Ha az OpenSSH-t Kerberos-szal szeretné használni, akkor néhány kezdeti beállításra van szükség.

Az alábbi lépések írják le az OpenSSH Kerberos támogatásának használatba vételéhez szükséges kezdeti beállítást.

1. Az OpenSSH klienseken és szervereken léteznie kell az /etc/krb5.conf fájlnak. Ez a fájl adja meg a Kerberosnak a használandó KDC-t, a jegyek élettartamát és egyéb beállításokat. A következő rész a krb5.conf fájlra mutat be egy példát:

```
[libdefaults]
ticket_lifetime = 600
default_realm = OPENSSH.AUSTIN.XYZ.COM
default_tkt_enctypes = des3-hmac-sha1 des-cbc-crc
default_tgs_enctypes = des3-hmac-sha1 des-cbc-crc

[realms]
OPENSSH.AUSTIN.xyz.COM = {
    kdc = kerberos.austin.xyz.com:88
    kdc = kerberos-1.austin.xyz.com:88
    kdc = kerberos-2.austin.xyz.com:88
    admin_server = kerberos.austin.xyz.com:749
    default_domain = austin.xyz.com
}

[domain_realm]
    .austin.xyz.com = OPENSSH.AUSTIN.XYZ.COM
    kdc.austin.xyz.com = OPENSSH.AUSTIN.XYZ.COM
```

2. Emellett fel kell venni az alábbi Kerberos szolgáltatásokat minden egyes kliens számítógép /etc/services fájljába:

```
kerberos      88/udp    kdc    # Kerberos V5 KDC
kerberos      88/tcp    kdc    # Kerberos V5 KDC
kerberos-adm  749/tcp           # Kerberos 5 admin/changepw
kerberos-adm  749/udp           # Kerberos 5 admin/changepw
krb5_prop     754/tcp           # Kerberos slave
                                #  propagation
```

3. Ha a KDC LDAP címtárat használ a felhasználói információk nyilvántartásaként, akkor olvassa el az “LDAP hitelesítési modul” oldalszám: 150 anyagot és a Kerberos kiadványokat. Emellett gondoskodjék a következő tevékenységek elvégzéséről:
   - Ahhoz, hogy a KDC használhassa az LDAP klienst, el kell indítani azt a **secldapclntd** paranccsal.
   - Az LDAP szerver démonnak (slapd) futnia kell.
4. Az OpenSSH szerveren adja hozzá az /etc/ssh/sshd_config fájlhoz a következő sorokat:

```
KerberosAuthentication yes
KerberosTicketCleanup yes
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
UseDNS yes
```

   Ha a UseDNS értéke **Yes**, akkor az ssh szerver fordított hoszt kikeresést végez a kapcsolódó kliens nevének megállapításához. Ez hoszt alapú hitelesítés esetén szükséges, vagy ha a legutolsó bejelentkezésre vonatkozó információk megjelenítésekor IP cím helyett hosztneveket szeretne használni.

   Megjegyzés: A fordított névkikeresések alatt beragadhatnak az ssh munkamenetek abban az esetben, ha a DNS szerverek nem elérhetőek. Ha ez megtörténik, akkor kihagyhatja a DNS kikeresési fázist a UseDNS paraméter no értékre állításával. Ha a UseDNS nincs beállítva explicit módon az /etc/ssh/sshd_config fájlban, akkor az alapértelmezett érték a UseDNS yes.
5. Az SSH szerveren futtassa a **startsrc -g ssh** parancsot az SSL szerver démon elindításához.
6. Az SSH kliens számítógépen a **kinit** paranccsal szerezze be a kezdeti meghatalmazást (vagyis jegymegadó jegyet). A TGT fogadását a **klist** paranccsal ellenőrizheti. A parancs a felhasználói azonosítójához tartozó összes meghatalmazást megjeleníti.
7. Csatlakozzon a szerverhez az **ssh** *felhasználó@szerver* paranccsal.
8. Ha a Kerberos megfelelően be van állítva a felhasználó hitelesítésére, akkor nem jelezik meg a jelszó beírására vonatkozó felszólítás, hanem automatikusan bejelentkezik az SSH szerverre.

A hálózat biztonságossá tétele

A következő fejezetek bemutatják az IP biztonsági szolgáltatás telepítését és beállítását, a szükséges és szükségtelen hálózati szolgáltatások felismerését és a hálózati biztonság megfigyelését.

TCP/IP biztonság

Ha telepítette az Átvitelvezérlési protokoll/Internet Protokoll (TCP/IP) és Hálózati fájlrendszer (NFS) szoftvert, akkor a rendszer beállítható hálózati kommunikációra.

A TCP/IP alapelveinek tárgyalása meghaladja jelen kézikönyv kereteit, ezért kénytelenek vagyunk a TCP/IP protokollal kapcsolatos biztonsági kérdésekre szorítkozni. További információk a TCP/IP telepítéséről és kezdeti beállításáról a *Networks and communication management* Átvitelvezérlési protokoll részében találhatók.

A rendszer adminisztrálását végző személynek egy sereg ok miatt teljesítenie kell bizonyos biztonsági szintet. A biztonsági szintet előírhatja például a vállalati házirend. Az is elképzelhető, hogy egy rendszer kormányzati rendszerekhez fér hozzá, amely megköveteli bizonyos biztonsági szint teljesítését. Ezek a biztonsági szabványok alkalmazhatók a hálózatra, az operációs rendszerre, az alkalmazásszoftverre, még a rendszeradminisztrátor által írt programokra is.

Ez a rész írja le a TCP/IP szabványos módban és biztonságos rendszereken rendelkezésre álló biztonsági szolgáltatásait, illetve itt tárgyaljuk a hálózati környezetek általános biztonsági szempontjait is.

TCP/IP és NFS szoftver telepítése után használja a Rendszergazdai kezelőfelület (SMIT) **tcpip** gyorselérését a rendszer konfigurálásához.

A **dacinet** parancsról további információkat az *Commands Reference* című kiadványban talál.

Operációs rendszer specifikus biztonság

A TCP/IP biztonsági szolgáltatásainak nagy része, mint például a hálózati hozzáférés felügyelet és a hálózatmegfigyelés, az operációs rendszer szolgáltatásain alapul.

A TCP/IP biztonságát a következő szakaszok körvonalazzák.

Hálózati hozzáférés felügyelet:

A hálózatkezelés biztonsági irányelvei az operációs rendszer biztonsági irányelveinek kiterjesztése, és a felhasználói hitelesítést, a kapcsolthitelesítést és az adatbiztonságot foglalja magában.

A következő fő összetevőből áll:

- A távoli hoszton biztosított Felhasználó hitelesítés ugyanúgy működik, mint amikor a felhasználó a helyi rendszerre jelentkezik be. A megbízható TCP/IP parancsok, például az **ftp**, a **rexec** és a **telnet** az operációs rendszer megbízható parancsaival azonos követelményeket támasztanak és azonos ellenőrzési folyamaton mennek keresztül.
- A Kapcsolat hitelesítés biztosítja, hogy a távoli hoszt a várt Internet protokoll (IP) címmel és névvel rendelkezik. Ez akadályozza meg, hogy egy hoszt egy másik hosztot játszhasson.
- Az Adatok importálási és exportálási biztonsága teszi lehetővé, hogy az adott biztonsági szintű adatok csak azonos biztonsági és hitelesítési szintű hálózati csatolókon folyhassanak. A szigorúan bizalmas adatok például csak szigorúan bizalmas minősítésű hálózati csatolók között folyhatnak.

Hálózatmegfigyelés:

A hálózatmegfigyelést a TCP/IP biztosítja, a megfigyelési alrendszer felhasználásával az alkalmazásprogramok megfigyelésére.

A megfigyelés célja a rendszer biztonságát érintő tevékenységek és az ezeket végző felhasználó feljegyzése.

A következő alkalmazásesemények kerülnek megfigyelésre:

- Hálózati hozzáférés
- Kapcsolat
- Adatok exportálása
- Adatok importálása

Az objektumok létrehozását és törlését az operációs rendszer figyeli meg. Az alkalmazás megfigyelés ilyen esetekben felfüggeszti majd újraindítja a megfigyelést a redundáns megfigyelés elkerülése érdekében.

Megbízható útvonal, megbízható parancsértelmező és biztonságos figyelmeztetés billentyű:

Az operációs rendszer a *megbízható útvonal* funkciót azért biztosítja, hogy megakadályozza a jogosulatlan programokat a felhasználói terminálok adatainak olvasásában. Ez az útvonal akkor kerül felhasználásra, ha biztonságos kommunikációs útvonalat kell kialakítani a rendszerrel, például a jelszavak cseréje vagy a rendszerbe bejelentkezés esetén.

Az operációs rendszer egy *megbízható parancsértelmezőt* ((**tsh**) is biztosít, amely csak olyan megbízható programokat futtat le, amelyek tesztelésre kerültek és biztonságosságuk igazolást nyert. A TCP/IP mindkét szolgáltatást támogatja a *biztonságos figyelmeztetés billentyű* (SAK) funkcióval együtt, amely kialakítja a felhasználó és a rendszer közötti biztonságos kommunikációhoz szükséges környezetet. A helyi SAK mindig elérhető, amikor a TCP/IP-t használja. A távoli SAK a **telnet** paranccsal érhető el.

A helyi SAK funkciója megegyezik a **telnet** használatakor és az operációs rendszer alkalmazásaiban: befejezi a **telnet** folyamatot, illetve a **telnet** parancsot futtató terminállal kapcsolatos összes további folyamatot. A telnet programon belül a **telnet send sak** parancsa segítségével kérhet megbízható útvonalat a távoli rendszerhez (**telnet** parancsmódban). A SAK kérés kezdeményezésére a **telnet set sak** parancsával beállíthat egy billentyűt.

A Megbízható számítástechnikai alapkörnyezetről további információkat a “Megbízható számítástechnikai alapkörnyezet” oldalszám: 1 szakaszból szerezhet.

TCP/IP parancs biztonsága

A TCP/IP bizonyos parancsai biztonságos környezetet nyújtanak működésük során. Ezek a parancsok a **ftp**, a **rexec** és a **telnet**.

Az **ftp** funkció biztonságos fájlátvitelt tesz lehetővé. A **rexec** parancs biztonságos környezetet nyújt a távoli hosztokon futtatott parancsokhoz. A **telnet** funkció biztonságos bejelentkezést tesz lehetővé a távoli hosztokra.

Az **ftp**, **rexec** és **telnet** parancs csak saját működésük során nyújt biztonságot. Ez azt jelenti, hogy más parancsok számára nem nyújtanak biztonságos környezetet. Ha a rendszert más műveletek számára is biztonságossá kívánja tenni, akkor használja a **securetcpip** parancsot. Ez a parancs lehetővé teszi a nem megbízható démonok és alkalmazások letiltását, illetve lehetőséget nyújt az IP réteg biztonságossá tételére is.

Az **ftp**, **rexec**, **securetcpip** és **telnet** parancsok a rendszer- és adatbiztonság alábbi formáit biztosítják:

ftp Az **ftp** parancs biztonságos környezetet nyújt a fájlok átviteléhez. Amikor egy felhasználó meghívja az **ftp** parancsot egy idegen hosztra, akkor a program bejelentkezési azonosítót kér. Egy alapértelmezett bejelentkezési név, a felhasználó aktuális helyi felhasználóneve megjelenik. A rendszer bekéri a távoli hoszton érvényes jelszót.

Az automatikus bejelentkezési folyamat végigkeresi a $HOME/.netrc fájlt, hogy van-e benne a távoli hoszton használható felhasználói azonosító és jelszó. Biztonsági okokból a $HOME/.netrc fájl engedélyeit 600-ra (tulajdonosi olvasás és írás) kell állítani. Ellenkező esetben az automatikus bejelentkezés meghiúsul.

Megjegyzés: Mivel a .netrc fájl a jelszavakat egy titkosítatlan fájlban tárolja, az **ftp** parancs automatikus bejelentkezés szolgáltatása nem áll rendelkezésre, ha a rendszer a **securetcpip** paranccsal került beállításra. A szolgáltatás újra engedélyezhető, ha a /etc/security/config fájl tcpip szakaszából eltávolítja az **ftp** parancsot.

A fájlátviteli funkció használatához az **ftp** parancs két TCP/IP kapcsolatot igényel, egyet a Fájlátviteli protokoll (FTP) számára, egyet az adatátvitelhez. A protokoll kapcsolata elsődleges, és azért biztonságos, mert megbízható hosztok között került kialakításra. A másodlagos kapcsolat az adatok tényleges átviteléhez szükséges, és a helyi és a távoli hoszt is ellenőrzi, hogy ugyanazzal a hoszttal került-e kialakításra, mint az elsődleges kapcsolat. Ha az elsődleges és másodlagos kapcsolat nem ugyanazzal a hoszttal létesült, akkor az **ftp** parancs hibaüzenetet jelenít meg, mely szerint az adatkapcsolat nem került hitelesítésre, ezek után kilép. A másodlagos kapcsolat ellenőrzése megakadályozza, hogy egy harmadik elfogja a nem neki szánt adatokat.

rexec A **rexec** parancs biztonságos környezetet nyújt a távoli hosztokon végrehajtott parancsokhoz. A felhasználónak bejelentkezési azonosítót és jelszót is meg kell adnia.

Az automatikus bejelentkezési szolgáltatás hatására a **rexec** parancs a helyi felhasználó $HOME/.netrc fájljában keres egy távoli hoszton használható felhasználói azonosítót és jelszót. Biztonsági okokból a $HOME/.netrc fájl engedélyeit 600-ra (tulajdonosi olvasás és írás) kell állítani. Ellenkező esetben az automatikus bejelentkezés meghiúsul.

Megjegyzés: Mivel a .netrc fájl a jelszavakat egy titkosítatlan fájlban tárolja, a **rexec** parancs automatikus bejelentkezési szolgáltatása nem áll rendelkezésre, ha a rendszer biztonságos módban működik. A szolgáltatás újbóli engedélyezéséhez a /etc/security/config fájl tcpip szakaszából távolítsa el a bejegyzést.

securetcpip
A **securetcpip** parancs engedélyezi a TCP/IP biztonsági szolgáltatásait. A parancs megszünteti a nem megbízható parancsok futtatásának lehetőségét. A **securetcpip** parancs futtatásakor a következő parancsok eltávolítására kerül sor:

- **rlogin** és **rlogind**
- **rcp**, **rsh**, és **rshd**
- **tftp** és **tftpd**
- **trpt**

A **securetcpip** paranccsal alakítható át a szabványos biztonsági szintet alkalmazó rendszer magasabb biztonsági szintűre. Átalakítás után a rendszeren nem kell újra kiadni a **securetcpip** parancsot, kivéve a TCP/IP újratelepítése esetén.

telnet vagy tn
A **telnet** (TELNET) parancs biztonságos környezetet nyújt a távoli hosztokra való bejelentkezéshez. A felhasználónak bejelentkezési azonosítót és jelszót is meg kell adnia. A rendszer a felhasználó terminálját a hoszthoz közvetlenül csatlakozó terminálként kezeli. Ez annyit tesz, hogy a terminál elérését engedélybitek határozzák meg. Más felhasználók (csoport és egyéb) nem rendelkeznek írási hozzáféréssel a terminálhoz, de írhatnak rá üzeneteket, amennyiben a tulajdonos ad nekik írási engedélyt. A **telnet** parancs emellett lehetővé teszi a távoli rendszer biztonságos héjának használatát a SAK segítségével. Ez a **telnet** parancsban beállítható billentyűsorrend eltér a helyi megbízható útvonalat meghívó sorrendtől.

Távoli parancs-végrehajtási hozzáférés:

Az /etc/hosts.equiv fájlban felsorolt hosztok felhasználói jelszó megadása nélkül futtathatnak bizonyos parancsokat a helyi rendszeren.

Az alábbi táblázat mutatja be, hogyan valósítható meg a távoli hosztok listázása, hozzáadása és eltávolítása a SMIT felületen vagy a parancssori felületen.

14. táblázat: Távoli parancsvégrehajtás hozzáférési feladatok

Feladat	SMIT gyorselérés	Parancs vagy fájl
Parancsvégrehajtási hozzáféréssel rendelkező hosztok listázása	**smit lshostsequiv**	/etc/hosts.equiv fájl megjelenítése
Parancsvégrehajtási hozzáférés megadása távoli hosztoknak	**smit mkhostsequiv**	/etc/hosts.equiv fájl szerkesztése[megjegyzés]
Parancsvégrehajtási hozzáférés megvonása távoli hosztoktól	**smit rmhostsequiv**	/etc/hosts.equiv fájl szerkesztése[megjegyzés]

Megjegyzés: A fájleljárásokkal kapcsolatos további információkat az *Files Reference* témakör "hosts.equiv fájlformátum TCP/IP-hez" része tartalmaz.

Tiltott fájlátviteli program felhasználók:

Az /etc/ftpusers fájlban felsorolt felhasználók távoli FTP hozzáférése nem megengedett. Tegyük fel például, hogy A felhasználó egy távoli rendszerre van bejelentkezve, és ismeri a helyi rendszer B felhasználójának jelszavát. Ha a B felhasználó meg van adva az /etc/ftpusers fájlban, akkor az A felhasználó még B jelszavának ismeretében sem tud FTP kapcsolatot kezdeményezni a B felhasználó fiókjával.

alábbi táblázat mutatja be, hogyan lehet végrehajtani a korlátozott felhasználók listázását, hozzáadását és eltávolítását SMIT felületen vagy a parancssorból.

Távoli FTP felhasználókkal kapcsolatos feladatok

Feladat	SMIT gyorselérés	Parancs vagy fájl
Korlátozott FTP felhasználók listázása	**smit lsftpusers**	/etc/ftpusers fájl megjelenítése
Korlátozott felhasználó hozzáadása	**smit mkftpusers**	/etc/ftpusers fájl szerkesztése[Megjegyzés]
Korlátozott felhasználó eltávolítása	**smit rmftpusers**	/etc/ftpusers fájl szerkesztése[Megjegyzés]

Megjegyzés: A fájleljárásokkal kapcsolatos további információkat az *Files Reference* témakör "ftpusers fájlformátum TCP/IP-hez" része tartalmaz.

Megbízható folyamatok

A megbízható program vagy megbízható folyamat olyan parancsfájl, démon vagy program, amely megfelel bizonyos biztonsági szabványoknak. Ezeket a biztonsági szabványokat az USA Hadügyminisztériuma állapítja meg, és ez végzi bizonyos megbízható programok minősítését is.

A megbízható programok megbízhatósága különböző szintekben állapítható meg. A biztonsági szintek az A1, B1, B2, B3, C1, C2 és D, ahol a legmagasabb biztonsági szintet az A1 jelenti. Minden biztonsági szintnek teljesítenie kell bizonyos követelményeket. A C2 szintű biztonság például a következő szabványokból áll:

program integritás
: Biztosítja, hogy a folyamat pontosan a szándéknak megfelelően működik.

modularitás
: A folyamat forráskódja olyan modulokra van bontva, amelyeket más modulok közvetlenül nem érhetnek el és nem befolyásolhatnak.

legkisebb jogosultság elve
: Kimondja, hogy a felhasználónak mindig a lehető legkisebb jogosultság mellett kell tevékenykednie. Ez azt jelenti, hogy ha egy felhasználónak egy adott fájlnak csak megjelenítésére van jogosultsága, akkor véletlenül sem fordulhat elő, hogy például módosítani is tudja a fájlt.

objektum újrafelhasználás korlátozása
: Többek között megakadályozza, hogy a felhasználó ne találhasson véletlenül olyan memóriaterületet, amely meg lett jelölve felülírásra, de még nem került törlésre, vagyis bizalmas információkat tartalmazhat.

A TCP/IP számos megbízható és sok megbízhatatlan démont tartalmaz.

Megbízható démonok például a következők:

- **ftpd**
- **rexecd**
- **telnetd**

Megbízhatatlan démonok például a következők:

- **rshd**
- **rlogind**
- **tftpd**

Ahhoz, hogy egy rendszer megbízható legyen, Megbízható számítástechnikai alapkörnyezetben kell működnie, vagyis önálló hosztként biztonságosnak kell lennie. Hálózatban az összes fájlszervernek, átjárónak, és többi hosztnak is megbízhatónak kell lennie.

Megbízható számítástechnikai alapkörnyezet

A Hálózati megbízható számítástechnikai alapkörnyezet (NTCB) hálózati biztonságot nyújtó hardverekből és szoftverekből áll. Ez a szakasz határozza meg az NTCB összetevőit, és ezeknek a TCP/IP-hez való viszonyát.

A hálózat hardveres biztonsági szolgáltatásait a TCP/IP-hez használt hálózati kártyák biztosítják. Ezek a kártyák felügyelik, hogy csak olyan adatokat vegyenek fel, amelyeket a helyi rendszerre címeztek, vagy üzenetszórással mindenkinek elküldtek.

Az NTCB szoftveres összetevői a megbízhatónak tekinthető programokból állnak. A biztonságos rendszer részét képező programokat és a hozzájuk tartozó fájlokat az alábbi táblázat sorolja fel könyvtáranként.

/etc könyvtár

Név	Tulajdonos	Csoport	Mód	Engedélyek
gated.conf	root	system	0664	rw-rw-r—
gateways	root	system	0664	rw-rw-r—
hosts	root	system	0664	rw-rw-r—
hosts.equiv	root	system	0664	rw-rw-r—
inetd.conf	root	system	0644	rw-r—r—
named.conf	root	system	0644	rw-r—r—
named.data	root	system	0664	rw-rw-r—
networks	root	system	0664	rw-rw-r—
protocols	root	system	0644	rw-r—r—
rc.tcpip	root	system	0774	rwxrwxr—
resolv.conf	root	system	0644	rw-rw-r—
services	root	system	0644	rw-r—r—
3270.keys	root	system	0664	rw-rw-r—
3270keys.rt	root	system	0664	rw-rw-r—

/usr/bin könyvtár

Név	Tulajdonos	Csoport	Mód	Engedélyek
host	root	system	4555	r-sr-xr-x
hostid	bin	bin	0555	r-xr-xr-x
hostname	bin	bin	0555	r-xr-xr-x
finger	root	system	0755	rwxr-xr-x
ftp	root	system	4555	r-sr-xr-x
netstat	root	bin	4555	r-sr-xr-x
rexec	root	bin	4555	r-sr-xr-x
ruptime	root	system	4555	r-sr-xr-x
rwho	root	system	4555	r-sr-xr-x
talk	bin	bin	0555	r-xr-xr-x
telnet	root	system	4555	r-sr-xr-x

/usr/sbin könyvtár

Név	Tulajdonos	Csoport	Mód	Engedélyek
arp	root	system	4555	r-sr-xr-x
fingerd	root	system	0554	r-xr-xr—
ftpd	root	system	4554	r-sr-xr—
gated	root	system	4554	r-sr-xr—
ifconfig	bin	bin	0555	r-xr-xr-x
inetd	root	system	4554	r-sr-xr—
named	root	system	4554	r-sr-x—
ping	root	system	4555	r-sr-xr-x
rexecd	root	system	4554	r-sr-xr—
route	root	system	4554	r-sr-xr—
routed	root	system	0554	r-xr-x—-
rwhod	root	system	4554	r-sr-xr—
securetcpip	root	system	0554	r-xr-xr—
setclock	root	system	4555	r-sr-xr-x
syslogd	root	system	0554	r-xr-xr—
talkd	root	system	4554	r-sr-xr—
telnetd	root	system	4554	r-sr-xr—

/usr/ucb könyvtár

Név	Tulajdonos	Csoport	Mód	Engedélyek
tn	root	system	4555	r-sr-xr-x

/var/spool/rwho könyvtár

Név	Tulajdonos	Csoport	Mód	Engedélyek
rwho (könyvtár)	root	system	0755	drwxr-xr-x

Adatbiztonság és információvédelem

A TCP/IP biztonsági szolgáltatása nem titkosítja a hálózaton továbbított felhasználói adatokat.

A jelszavak és egyéb érzékeny adatok potenciális kiszivárgásához vezető kommunikációs kockázatok felmérése, és a kockázat mértéke alapján a megfelelő óvintézkedések megtétele.

A TCP/IP biztonsági szolgáltatás DoD környezetben használata megkövetelheti a DOD 5200.5 és NCSD-11 kommunikációs biztonsági minősítést.

Felhasználó alapú TCP port hozzáférés felügyelet az internet portok kizárólagos hozzáférés felügyeletével

Az Internet portok kizárólagos hozzáférés felügyelete (DACinet) lehetővé teszi az AIX hosztok közötti kommunikációhoz használt TCP portok felhasználó alapú hozzáférés felügyeletét.

Az AIX rendszer használhat egy további TCP fejlécet rendszerek között felhasználói és csoport információk szállítására. A DACinet szolgáltatás lehetővé teszi a célrendszer adminisztrátora számára, hogy hozzáférés felügyeletet vezessen be a célport, a kezdeményező felhasználói azonosító és hoszt alapján.

Emellett a DACinet segítségével az adminisztrátor korlátozhatja a helyi portokat csak root használatra. UNIX rendszerek - például az AIX - az 1024 alatti portokat privilegizált portnak tekintik, amelyeket csak a root nyithat meg. Az AIX lehetővé teszi, hogy 1024 felett is megjelölhessen olyan portokat, amelyeket csak a root nyithat meg, vagyis megakadályozhatja, hogy a felhasználók közismert portokon szervereket üzemeltessenek.

A beállításoktól függően a DACinet funkcionalitást nem biztosító rendszerek elképzelhető, hogy nem tudnak csatlakozni a DACinet rendszerekhez. A hozzáférés ilyenkor még a DACinet kezdeti szakaszában visszautasításra kerül. A DACinet az engedélyezés után nem tiltható le.

A **dacinet** parancs elfogad hosztnévként megadott címeket, pontozott decimális hoszt címeket és olyan hálózati címeket, amelyek után meg van adva a hálózati előtag hossza.

A következő példa egyetlen hosztot ad meg, amely a *host.domain.org* teljes képzésű hosztnéven ismert:

```
host.domain.org
```

A következő példa egy olyan egyedülálló hosztot ad meg, amely a 10.0.0.1 IP címe alapján ismert.

```
10.0.0.1
```

A következő példa azt a teljes hálózatot adja meg, amelynél az első 24 bit (a hálózati előtag hossza) 10.0.0.0:

```
10.0.0.0/24
```

Ebbe a hálózatba a 10.0.0.1 és 10.0.0.254 közötti összes IP cím beletartozik.

TCP alapú szolgáltatások hozzáférés felügyelete:

A DACinet az /etc/rc.dacinet indítási fájlt, és az /etc/security/priv, /etc/security/services illetve /etc/security/acl konfigurációs fájlokat használja.

Az /etc/security/services fájlban felsorolt portokat úgy tekinti, hogy mentesek az ACL ellenőrzésektől. A fájl formátuma az /etc/services fájlt követi. Inicializálásának legegyszerűbb módja, ha átmásolja a fájlt az /etc könyvtárból az /etc/security könyvtárba, majd törli belőle az összes olyan portot, amelyen nem kíván ACL ellenőrzést

végezni. Az ACL-ek két helyen kerülnek tárolásra. A pillanatnyilag aktív ACL a kernelben van, és a dacinet aclls paranccsal futtatható. Az /etc/rc.tcpip parancs által a következő rendszertöltéskor aktivált ACL-ek az /etc/security/acl fájlban találhatók. Formátuma a következő:

```
szolgáltatás hoszt/előtag [felhasználó|csoport]
```

Ahol a szolgáltatás egy számmal megadott szolgáltatás vagy az /etc/services egyik bejegyzése lehet, a hoszt megadható hosztnévként vagy alhálózati maszk meghatározással ellátott hálózati címként, a felhasználót vagy csoportot pedig az u: vagy g: előtag jelöli. Ha nincs megadott felhasználó vagy csoport, akkor az ACL csak a küldő hosztot veszi számításba. A szolgáltatás - előjele kifejezetten letiltja a hozzáférést. Az ACL-ek kiértékelése első megfeleléssel történik. Ily módon megadható egy felhasználói csoport hozzáférése, és letiltható egy adott felhasználó a rá vonatkozó ACL-nek a csoport ACL elé helyezésével.

Az /etc/services fájl tartalmaz két olyan bejegyzést, amelyek portszámát az AIX nem támogatja. A rendszeradminisztrátornak el kell távolítania mindkét sort az **mkCCadmin** parancs futtatása előtt. Távolítsa el az /etc/services fájl következő sorait:

```
sco_printer     70000/tcp     sco_spooler    # For System V print IPC
sco_s5_port     70001/tcp     lpNet_s5_port  # For future use
```

DACinet használati példák:

Például amikor a DACinet szolgáltatással korlátozza a TCP/25 bejövő port elérését a root felhasználóra, akkor csak a root felhasználók érhetik el ezt a portot más AIX hosztokról, így korlátozva annak lehetőségét, hogy normál felhasználók meghamisítsanak e-maileket pusztán telnetezéssel az áldozat TCP/25 portjára.

A következő példa azt mutatja be, hogyan állítható be az X protokoll (X11) csak root hozzáférésre. Győződjön meg róla, hogy az /etc/security/services fájl X11 bejegyzése eltávolításra került, vagyis a szolgáltatásra nem vonatkoznak ACL-ek.

Feltéve, hogy az összes csatlakozó rendszer a 10.1.1.0/24 alhálózaton található, az X (TCP/6000) portról a root felhasználó kivételével mindenkit kitiltó ACL bejegyzés a következőképpen nézne ki az /etc/security/acl fájlban:

```
      6000    10.1.1.0/24 u:root
```

Ha a Telnet szolgáltatást a friends csoport felhasználóira szeretné korlátozni a forrásrendszertől függetlenül, akkor távolítsa el a telnet bejegyzését az /etc/security/services fájlból, majd használja a következő ACL bejegyzést:

```
telnet    0.0.0.0/0   g:friends
```

A webszerver használatának megtiltása Fred számára, miközben mindenki más használhatja:

```
-80    0.0.0.0/0 u:fred
80     0.0.0.0/0
```

Helyi szolgáltatások futtatása privilegizált portokon:

Ha meg kívánja akadályozni, hogy a normál felhasználók szerverek futtathassanak bizonyos portokon, akkor ezek megjelölhetők privilegizált portként.

Normálisan minden felhasználó megnyithat 1024 feletti portokat. A felhasználók például elhelyezhetnek egy szervert a 8080-as vagy 1080-as porton, amelyet általában web proxy szerverek illetve a SOCKS szerverek használnak. A **dacinet setpriv** parancs használható privilegizált portok megjelölésére a futó rendszeren. A rendszer indításakor privilegizáltnak megjelölt portokat az /etc/security/priv fájlban kell megadni.

A portok a fájlban az /etc/services fájlban megadott szimbolikus nevükkel és portszámukkal is felsorolhatók. A következő bejegyzés megakadályozza, hogy a nem root felhasználók SOCKS vagy Lotus Notes szervereket futtassanak a szokásos portokon:

```
1080
lotusnote
```

Megjegyzés: A szolgáltatás nem akadályozza meg a felhasználót a program futtatásában. Ez csak azt akadályozza meg, hogy a felhasználó azon a közismert porton futtassa a szolgáltatásokat, amelyet azok általában használni szoktak.

Hálózati szolgáltatások

Nyitott kommunikációs portokkal rendelkező hálózati szolgáltatások azonosítása és biztonságossá tétele.

Portok használata

AZ alábbi táblázat bemutatja az ismert porthasználatot az AIX operációs rendszeren.

Megjegyzés: Ez a lista több AIX rendszer különféle telepített szoftverkonfigurációinak áttekintésével lett kialakítva.

Lehetséges, hogy az alábbi lista nem tartalmazza az AIX operációs rendszeren létező összes szoftverre vonatkozó porthasználatot:

Port/protokoll	Szolgáltatásnév	Álnevek
13/tcp	daytime	Daytime (RFC 867)
13/udp	daytime	Daytime (RFC 867)
21/tcp	ftp	File Transfer [Control]
21/udp	ftp	File Transfer [Control]
23/udp	telnet	Telnet
23/udp	telnet	Telnet
25/tcp	smtp	Simple Mail Transfer
25/udp	smtp	Simple Mail Transfer
37/tcp	time	Time
37/udp	time	Time
111/tcp	sunrpc	SUN Remote Procedure Call
111/udp	sunrpc	SUN Remote Procedure Call
161/tcp	snmp	SNMP
161/udp	snmp	SNMP
199/tcp	smux	SMUX
199/udp	smux	SMUX
512/tcp	exec	remote process execution;
513/tcp	login	remote login a la telnet;
514/tcp	shell	cmd
514/udp	syslog	Syslog
518/tcp	ntalk	Talk
518/udp	ntalk	Talk
657/tcp	rmc	RMC
657/udp	rmc	RMC
1334/tcp	writesrv	writesrv
1334/udp	writesrv	writesrv
2279/tcp	xmquery	xmquery
2279/udp	xmquery	xmquery
32768/tcp	filenet-tms	FileNet TMS
32768/udp	filenet-tms	FileNet TMS
32769/tcp	filenet-rpc	FileNet RPC
32769/udp	filenet-rpc	FileNet RPC
32770/tcp	filenet-nch	FileNet NCH
32770/udp	filenet-nch	FileNet NCH

Port/protokoll	Szolgáltatásnév	Álnevek
32771/tcp	filenet-rmi	FileNet RMI
32771/udp	filenet-rmi	FileNet RMI
32772/tcp	filenet-pa	FileNet Process Analyzer
32772/udp	filenet-pa	FileNet Process Analyzer
32773/tcp	filenet-cm	FileNet Component Manager
32773/udp	filenet-cm	FileNet Component Manager
32774/tcp	filenet-re	FileNet Rules Engine
32774/udp	filenet-re FileNET Rules Engine	FileNet Rules Engine
32775/tcp	filenet-pch	Performance Clearinghouse
32775/udp	filenet-pch	Performance Clearinghouse
32776/tcp	filenet-peior	FileNet BPM IOR
32776/udp	filenet-peior	FileNet BPM IOR
32777/tcp	filenet-obrok	FileNet BPM CORBA
32777/udp	filenet-obrok	FileNet BPM CORBA

Hálózati szolgáltatások azonosítása nyílt kommunikációs portokkal

A kliens/szerver alkalmazások kommunikációs portokat nyitnak meg a szerveren, így lehetővé téve az alkalmazásoknak a bejövő kliens kérések fogadását.

Mivel a nyitott portok biztonsági támadások lehetséges célpontjai lehetnek, azonosítani kell a nyitott portokkal rendelkező alkalmazásokat, és be kell zárni a szükségtelenül megnyitott portokat. Ez a gyakorlat azért hasznos, mert segít tudatosítani, hogy a nyilvános rendszerek az Internet valamennyi felhasználója számára elérhetők.

A nyitott portok meghatározásához tegye a következőket:

1. Azonosítsa a szolgáltatásokat a **netstat** paranccsal az alábbiak szerint:

   ```
   # netstat -af inet
   ```

   A parancs kimenetére az alábbiakban láthat egy példát. A **netstat** parancs kimenetének utolsó oszlopa jelzi az egyes szolgáltatások állapotát. A bejövő kapcsolatokra várakozó szolgáltatások állapota FIGYELÉS.

Ez egy példa a parancs kimenetére a **netstat** parancs futtatásakor.

Aktív Internet kapcsolat (szervereket is beleértve)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
tcp4	0	0	*.echo	*.*	LISTEN
tcp4	0	0	*.discard	*.*	LISTEN
tcp4	0	0	*.daytime	*.*	LISTEN
tcp	0	0	*.chargen	*.*	LISTEN
tcp	0	0	*.ftp	*.*	LISTEN
tcp4	0	0	*.telnet	*.*	LISTEN
tcp4	0	0	*.smtp	*.*	LISTEN
tcp4	0	0	*.time	*.*	LISTEN
tcp4	0	0	*.www	*.*	LISTEN
tcp4	0	0	*.sunrpc	*.*	LISTEN
tcp	0	0	*.smux	*.*	LISTEN
tcp	0	0	*.exec	*.*	LISTEN
tcp	0	0	*.login	*.*	LISTEN

Ez egy példa a parancs kimenetére a **netstat** parancs futtatásakor.

Aktív Internet kapcsolat (szervereket is beleértve)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
tcp4	0	0	*.shell	*.*	LISTEN
tcp4	0	0	*.klogin	*.*	LISTEN
udp4	0	0	*.kshell	*.*	LISTEN
udp4	0	0	*.echo	*.*	
udp4	0	0	*.discard	*.*	
udp4	0	0	*.daytime	*.*	
udp4	0	0	*.chargen	*.*	
udp4	0	0	*.time	*.*	
udp4	0	0	*.bootpc	*.*	
udp4	0	0	*.sunrpc	*.*	
udp4	0	0	255.255.255.255.ntp	*.*	
udp4	0	0	1.23.123.234.ntp	*.*	
udp4	0	0	localhost.domain.ntp	*.*	
udp4	0	0	name.domain..ntp	*.*	

.....

2. Az operációs rendszer szolgáltatásai és a portok közötti leképezés meghatározásához nyissa meg az /etc/services fájlt, és nézze meg az Internet hozzárendelt számainak hatósága (IANA) szakaszt.

   A következő a /etc/services fájl példatöredéke:

```
tcpmux  1/tcp  # TCP Port Service Multiplexer
tcpmux  1/tcp  # TCP Port Service Multiplexer
Compressnet  2/tcp  # Management Utility
Compressnet  2/udp  # Management Utility
Compressnet  3/tcp  # Compression Process
Compressnet  3/udp  Compression Process
Echo  7/tcp
Echo  7/udp
discard  9/tcp  sink null
discard  9/udp  sink null
..............
rfe  5002/tcp  # Radio Free Ethernet
rfe  5002/udp  # Radio Free Ethernet
rmonitor_secure  5145/tcp
rmonitor_secure  5145/udp
pad12sim  5236/tcp
pad12sim  5236/udp
sub-process  6111/tcp  # HP SoftBench Sub-Process Cntl.
sub-process  6111/udp  # HP SoftBench Sub-Process Cntl.
xdsxdm  6558/ucp
xdsxdm  6558/tcp
afs3-fileserver  7000/tcp  # File Server Itself
afs3-fileserver  7000/udp  # File Server Itself
af3-callback  7001/tcp  # Callbacks to Cache Managers
af3-callback  7001/udp  # Callbacks to Cache Managers
```

3. Zárja be a szükségtelen portokat a futó szolgáltatások eltávolításával.

Megjegyzés: A 657-es portot az Erőforrásfigyelő és vezérlő (RMC) használja a csomópontok közötti kommunikációhoz. Ezt a portot nem tilthatja le és egyéb módon sem korlátozhatja.

TCP és UDP socketek azonosítása

Adatok érkezésére várakozó FIGYELÉS állapotú TCP illetve tétlen UDP socketek azonosításához használja a **netstat -af** parancs egyik változatát, az **lsof** parancsot.

A FIGYELÉS állapotú TCP socketek és a tétlen UDP socketek megjelenítéséhez például futtassa az **lsof** parancsot az alábbiak szerint:

```
# lsof -i | egrep "COMMAND|LISTEN|UDP"
```

A kapott kimenet a következőhöz fog hasonlítani:

Command	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
dtlogin	2122	root	5u	IPv4	0x70053c00	0t0	UDP	*:xdmcp
dtlogin	2122	root	6u	IPv4	0x70054adc	0t0	TCP	*:32768(LISTEN)
syslogd	2730	root	4u	IPv4	0x70053600	0t0	UDP	*:syslog
X	2880	root	6u	IPv4	0x70054adc	0t0	TCP	*:32768(LISTEN)
X	2880	root	8u	IPv4	0x700546dc	0t0	TCP	*:6000(LISTEN)
dtlogin	3882	root	6u	IPv4	0x70054adc	0t0	TCP	*:32768(LISTEN)
dtgreet	4656	root	6u	IPv4	0x70054adc	0t0	TCP	*:32768(LISTEN)

A folyamatazonosító azonosítása után a programról a következő paranccsal kaphat további információkat:

```
" # ps -fp PID#"
```

A kimenet tartalmazza a parancs elérési útját és parancsnevét, amellyel ezután megtekintheti a program man oldalát.

Internet protokoll biztonság

Az IP biztonság az IP réteg adatforgalmának titkosításával lehetővé teszi a biztonságos kommunikációt az Interneten és a vállalati hálózaton.

IP biztonság áttekintése

Az IP biztonság lehetővé teszi a felhasználóknak, hogy az alkalmazások adatforgalmát az alkalmazások módosítása nélkül titkosítsák. Másképp fogalmazva tetszőleges adatátvitel titkosítható.

IP biztonság és az operációs rendszer:

Az operációs rendszer az IETF által kifejlesztett nyílt, szabványos biztonsági technológiát, az IP biztonsági szolgáltatást (IPSec) használja.

Az IP biztonság a kommunikációs verem IP rétegében biztosítja a teljes adatforgalom kriptográfiai védelmét. A meglévő alkalmazások ily módon nem szorulnak módosításra. Az IP biztonsági protokoll olyan ipari szabvány hálózatbiztonsági keretrendszer, amelyet az IETF az IPv4 és az IPv6 környezetekre is leírt.

Az IP biztonsági protokoll a következő kriptográfiai eljárásokkal biztosítja az adatforgalom védelmét:

Hitelesítés
: Egy hoszt vagy végpont azonosságának ellenőrzésére szolgáló folyamat

Integritásellenőrzés
: A hálózaton átvitt adatok módosítását kizáró folyamat

Titkosítás
: A bizalmasság megőrzésére szolgáló folyamat, amely "elrejti" a hálózaton átküldött adatokat és magán IP címeket

A hitelesítési algoritmusok a küldő azonosságát és az adatok integritását kriptográfiai kivonatkészítési funkcióval biztosítják, amely a (rögzített IP fejléc mezőkkel együtt vett) adatcsomagból egy titkos kulccsal egyedi kivonatot készítenek. A fogadó oldalán az az adatokat ugyanaz a funkció és kulcs dolgozza fel. Ha az adatok megváltoztak vagy a küldő kulcsa érvénytelen, akkor az adatcsomagot a rendszer eldobja.

A titkosítás egy kriptográfiai algoritmus és egy kulcs felhasználásával úgy módosítja az adatokat, hogy az ezek értelmezhetetlenek legyenek a kulcs nélkül; ezt néha *rejtjelszövegnek* is hívjuk. A titkosított adatok nem olvashatók az átvitel során. Megérkezése után a rendszer ugyanazon algoritmus és kulcs alkalmazásával visszaállítja az adatokat. A titkosításnak a hitelesítéssel együtt kell történnie, hogy a titkosított adatok integritása is ellenőrizhető legyen.

Ezeket az alapvető szolgáltatásokat az IP biztonsági protokoll a Beágyazott biztonsági kiterjesztés (ESP) és a Hitelesítési fejléc (AH) protokollokkal valósítja meg. Az ESP biztosítja a bizalmasságot az eredeti IP csomag titkosításával, egy ESP fejléc összeállításával és a rejtjelszövegnek az ESP csomagba helyezésével.

Az AH önmagában is használható hitelesítésre és integritás ellenőrzésre, ha a bizalmasság nem kérdés. AH esetén az IP fejléc statikus mezőin és adatain kerül végrehajtásra egy kivonatkészítési algoritmus, amelynek eredménye egy kulcsolt kivonat. A fogadó a saját kulcsával szintén kiszámítja a kivonatot, és összehasonlítja a csomagban kapottal, így ellenőrizve a küldő azonosságát, és azt, hogy a csomag tartalma nem változott meg.

IP biztonsági szolgáltatások:

Az alábbiakban az IP biztonság szolgáltatásait találja.

Az alábbi szolgáltatások érhetők el Internet kulcscserével AIX operációs rendszerhez:

- Támogatja az AES 128 bites, 192 bites és 256 bites algoritmusokat.
- Hardveres gyorsítás a 10/100 Mbps Ethernet PCI Adapter II csatolóval.
- AH támogatás az RFC 2402, illetve ESP támogatás az RFC 2406 alapján.
- Lehetőség van kézi alagutak beállítására IPv6 alagutak esetén, illetve az olyan rendszerekkel való együttműködéshez, amelyek nem támogatják az automatikus IKE kulcsfrissítési módszert.
- Alagút és szállítási módú beágyazás hoszt és átjáró végponttal rendelkező alagutakhoz.
- HMAC (Kivonat alapú üzenethitelesítési kód) Üzenet kivonat 5 (MD5) és HMAC SHA (Biztonságos kivonatkészítési algoritmus) hitelesítési algoritmusok.
- A titkosítási algoritmusok között megtalálható a 64 bites kezdeti vektorral (IV) rendelkező 56 bites DES-CBC, a tripla DES, a DES-CBC 4 (32 bites IV) és az AES CBC.
- Kettős IP verem támogatás (IPv4 és IPv6).
- Az IPv4 és IPv6 forgalom is beágyazható és szűrhető. Mivel a vermek önállóak, az IP biztonsági funkció is függetlenül állítható be minden veremnél.
- Biztonságos és nem biztonságos forgalom szűrése többféle IP jellemző, például forrás- és célcím, csatoló, protokoll, portszám és egyebek alapján.
- A legtöbb alagúttípusnál lehetőség van a szűrőszabályok automatikus létrehozására.
- Hosztnevek használatának lehetősége célcímként alagutak és szűrőszabályok meghatározásakor. A hosztneveket a rendszer automatikusan IP címre alakítja át (ha DNS elérhető).
- IP biztonsági események naplózása a **syslog** naplóba.
- Rendszer nyomkövetések és statisztikák használata a hibafelderítésben.
- A felhasználó által megadott alapértelmezett tevékenységgel a felhasználó megadhatja a meghatározott alagutaknak nem megfelelő forgalom engedélyezését is.

AIX 6.1 TL 05 vagy újabb rendszeren az Internet kulcscsere protokollal a következő további szolgáltatások érhetők el:

- IPSec támogatás az RFC 4301, AH támogatás az RFC 4302 és ESP támogatás az RFC 4303 alapján.
- Rejtjel-alapú üzenethitelesítési kód (CMAC) AES XCBC hitelesítési algoritmusai
- Titkosítási algoritmusok: AES 128 bites, 192 bites, 256 bites GCM (16 bites IV), AES-128-GMAC, AES-192-GMAC és AES-256-GMAC

- Szűrőútvonalak porttartománya
- Kiterjesztett sorozatszámok

Internet kulcscsere szolgáltatások:

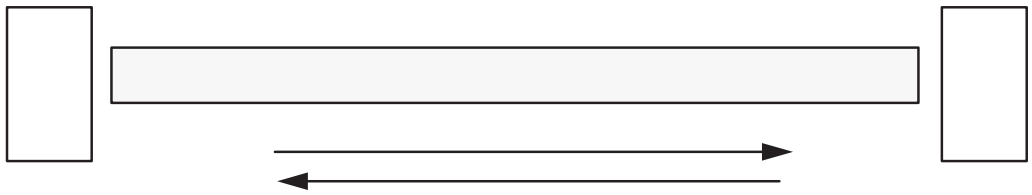
A következő szolgáltatások érhetők el az Internet kulcscsere (IKE) protokollal AIX rendszeren.

AIX 6.1 vagy újabb rendszeren az Internet kulcscsere protokollal a következő további szolgáltatások érhetők el:
- AH támogatás HMAC SHA2 256 bites kivonathoz (TL 04 vagy újabb).
- ESP titkosítási támogatás GCM AES 128 bit, 192 bit, 256 bit (16 bites IV), GMAC AES 128 bites, 192 bites, 256 bites algoritmusokkal; ESP hitelesítési támogatás HMAC MD5 és HMAC SHA1 titkosítással (TL 04 vagy újabb).
- IKEv1 (RFC2409) és IKEv2 (RFC4306) támogatott (TL 02 vagy újabb). Az IKEv1-et az **isakmpd** démon, az IKEv2-et pedig az **ikev2d** démon támogatja (TL 02 vagy újabb). Az IKEv1 és az IKEv2 csatorna együtt is létezhet.
- CMAC_AES_XCBC és HMAC_SHA2_256 integritási algoritmusok támogatása (TL 04 vagy újabb).
- PRF_SHA2_256 PRF algoritmus támogatása (TL 04 vagy újabb).
- 14, 19 és 24 Diffie Hellman csoport támogatása (TL 04 vagy újabb).

Biztonsági megegyezések:

A biztonságos kommunikáció alapját képező alapelvet *biztonsági megegyezésnek* hívjuk. A biztonsági megegyezések kapcsolják össze a biztonsági paraméterek egy halmazát egy forgalomtípussal.

Az IP biztonság által védett adatok esetén külön biztonsági megegyezés tartozik minden egyes irányhoz és fejléctípushoz. A biztonsági megegyezésekben tárolt információk magukban foglalják a kommunikáló felek IP címeit, a biztonsági paraméter indexnek nevezett egyedi azonosítót, a hitelesítéshez és titkosításhoz használt algoritmust, a hitelesítési és titkosítási kulcsokat, illetve ezen kulcsok élettartamát. A következő ábra az A és B hoszt közötti biztonsági megegyezéseket mutatja be.



6. ábra: Biztonságos alagút kialakítása az A és B hoszt között

Az ábra egy virtuális alagutat mutat be az A és B hoszt között. A B biztonsági megegyezés a B hosztból az A hosztba mutató nyíl. A biztonsági megegyezések egy célcímet, biztonsági paraméter indexet, kulcsot, titkosítási algoritmust és formátumot, hitelesítési algoritmust és kulcs élettartamot adnak meg.

A kulcskezelés célja az IP forgalmat védő biztonsági megegyezések egyeztetése és kiszámítása.

Alagutak és kulcskezelés:

Egy alagút segítségével egyeztetheti és kezelheti a biztonsági megegyezéseket, amelyek a két hoszt közötti biztonságos kommunikáció beállításához szükségesek.

A rendszer az alábbi típusú alagutakat támogatja; ezek mindegyike eltérő kulcskezelési technikát alkalmaz:

- IKE alagutak (dinamikusan változó kulcsok, IETF szabvány)
- Kézi alagutak (statikus, állandó kulcsok, IETF szabvány)

Internet kulcscsere alagút támogatása:

Az IKE alagutak az IETF által fejlesztett Internet biztonsági megegyezés és kulcskezelési protokoll (ISAKMP)/Oakley szabványokon alapulnak. Ez a protokoll lehetővé teszi a biztonsági paraméterek egyeztetését és frissítését, illetve a kulcsok biztonságos cseréjét.

A következő típusú hitelesítések támogatottak:

- Előzetesen megosztott kulcs.
- X.509v3 digitális igazolás aláírások.
- AIX 6.1 TL 04 vagy újabb rendszeren az IKEv2 támogatja az ECDSA-256 digitális igazolás aláírásokat a digitális igazolásokra alapozott X509v3 hitelesítési módszer részeként.

Az egyeztetés kétfázisos megközelítést alkalmaz. Az első fázis hitelesíti a kommunikáló feleket és megadja a 2. fázisban folytatott biztonságos kommunikációhoz használt algoritmusokat. A 2. fázisban történik az adatforgalom során használt IP biztonsági paraméterek egyeztetése, illetve a kulcsok létrehozása és cseréje.

Az alábbi táblázat mutatja be az IKE alagutak AH és ESP protokolljaival használható hitelesítési algoritmusokat.

15. táblázat: Hitelesítési algoritmusok IKE alagút támogatáshoz

Algoritmus	AH IPv4 és IPv6	ESP IPv4 és IPv6
HMAC MD5	X	X
HMAC SHA1	X	X
DES CBC 8		X
Tripla DES CBC		X
AES CBC (128, 192, 256)		X
ESP Null		X
AES-XCBC-MAC-96	X	X
AES GCM (128, 192, 256)		X
AES GMAC (128, 192, 256)	X	
ESP_ENCR_NULL_ AUTH_AES_GMAC		X

Kézi alagúttámogatás:

A kézi alagutak visszamenőleges kompatibilitást biztosítanak az olyan gépekkel, amelyek nem támogatják az IKE kulcskezelési protokollokat. A kézi alagutak hátránya, hogy a kulcs értékek statikusak. A titkosítási és hitelesítési kulcsok azonosak az alagút élettartama során, és saját kezűleg kell azokat frissíteni.

Az alábbi táblázat mutatja be a kézi alagutak AH és ESP protokolljaival használható hitelesítési algoritmusokat.

Algoritmus	AH IPv4	AH IPv6	ESP IPv4	ESP IPv6
HMAC MD5	X	X	X	X
HMAC SHA1	X	X	X	X
AES CBC (128, 192, 256)			X	X
Tripla DES CBC			X	X
DES CBC 8			X	X
DES CBC 4			X	X

Mivel az IKE alagutak hatékonyabb biztonságot nyújtanak, az IKE az ajánlott kulcskezelési módszer.

Natív szűrési képesség:

A *szűrés* egy olyan alapvető funkció, amely lehetővé teszi, hogy a bejövő és kimenő csomagokat a rendszer különféle jellemzők alapján elutasítsa vagy elfogadja. Ezzel a felhasználók vagy rendszeradminisztrátorok felügyelhetik a hoszt és más hosztok közötti forgalmat.

A szűrés különféle csomagtulajdonságok, például forrás- és célcím, IP verziószám, alhálózati maszk, protokoll, port, útválasztási jellemzők, töredezettség, csatoló és alagút meghatározás alapján történhet.

A *szűrőszabályoknak* is nevezett szabályok társítják a különféle forgalmat egy adott alagúthoz. Kézi alagutakat alkalmazó alapszintű konfiguráció esetén, amikor egy felhasználó hoszt-hoszt alagutat határoz meg, akkor a rendszer automatikusan előállítja azokat a szűrőszabályokat, amelyek a hoszt forgalmát a védett alagúton továbbítják. Egyedibb forgalmi igények, például alhálózatok közötti alagutak esetén a szűrőszabályok módosíthatók és lecserélhetők, így biztosítva lehetőséget az adott alagút forgalmának precízebb felügyeletére.

IKE alagutak esetén a szűrőszabályok szintén automatikusan kerülnek előállításra és beszúrásra a szűrők táblájába az alagút aktiválása után.

Hasonlóan, az alagút módosításakor vagy törlésekor az alagúthoz tartozó szabályok automatikusan törlődnek, így megkönnyítve az IP biztonság konfigurációjának karbantartását, és lecsökkentve a felhasználói hibák esélyét. Az importálás és exportálás segédprogramokkal az alagút meghatározások terjeszthetők és megoszthatók több számítógép és a tűzfalak között, amely nagy számú számítógép esetén jelentősen megkönnyíti az adminisztrációt.

A szűrőszabályok társítják az adott forgalomtípust egy alagúthoz, de a szűrt adatoknak nem feltétlenül kell egy alagúton áthaladniuk. A szűrőszabályok ilyetén felhasználásával az operációs rendszer biztosíthat alapszintű tűzfal tevékenységeket az olyan esetekben, amikor valaki korlátozni kívánja a számítógép forgalmát egy intraneten vagy valódi tűzfallal nem rendelkező hálózaton. Ebben az esetben a szűrőszabályok egy másodlagos védelmet jelentenek a számítógép számára.

A szűrőszabályokat az előállításuk után a rendszer egy táblában tárolja, amelyet betölt a kernelbe. Amikor a csomagok készen állnak a küldésre vagy fogadásra, akkor a rendszer sorban ellenőrzi a szűrőszabályokat annak megállapításához, hogy a csomag engedélyezendő, eldobandó vagy alagúton keresztül továbbítandó. A szabályok keresése egy megfelelő szabály megtalálásáig vagy az alapértelmezett szabály eléréséig tart.

Az IP biztonság funkció emellett lehetővé teszi a nem biztonságos csomagok felhasználó által megadott feltételek szerinti szűrését, amely biztosítja az IP biztonság hitelesítési és titkosítási szolgáltatásait nem igénylő hálózatokkal és számítógépekkel folytatott IP forgalom felügyeletét is.

Digitális igazolás támogatása:

Az IP biztonság támogatja az X.509 digitális igazolások használatát.

Az igazolási kérések kezelését, a kulcsadatbázis karbantartását és a további adminisztrációs funkciókat a Kulcskezelési eszköz végzi.

A digitális igazolások leírását a Digitális igazolások beállítása helyen találja. A kulcskezelő és funkcióinak leírását az IBM kulcskezelő eszköz használata rész tartalmazza.

Virtuális magánhálózatok és IP biztonság:

A virtuális magánhálózatok (VPN) lehetővé teszik a belső intranet kiterjesztését nyilvános hálózatok, például az Internet felett.

A virtuális magánhálózatok lényegében privát alagúton szállítanak információkat az Interneten keresztül a távoli felhasználók, telephelyek és üzleti partnerek/szállítók között. A vállalatok igénybe vehetnek egy Internet szolgáltatót, amelynek segítségével helyi telefonszámon juthatnak költséghatékony Internet hozzáféréshez, így kiküszöbölve a bérelt vonalak, távolsági hívások és hasonló megoldások szükségességét. A VPN megoldások használhatják az IPSec biztonsági szabványt, mivel az IPSec az IETF által választott ipari szabvány hálózati biztonsági keretrendszer, amely IPv4 és IPv6 környezetben is használható, és a meglévő alkalmazások módosítását sem igényli.

Az AIX operációs rendszer virtuális magánhálózatainak tervezéséhez és megvalósításához ajánlott irodalom az *A Comprehensive Guide to Virtual Private Networks, Volume III: Cross-Platform Key and Policy Management*, ISBN SG24-5309-00 című kiadvány 9. fejezete. A kézikönyv a http://www.redbooks.ibm.com/redbooks/SG245309.html Internetcímen is elérhető.

IP biztonsági szolgáltatás telepítése

Az AIX IP biztonsági szolgáltatása külön telepíthető és tölthető be.

A telepítendő fájlkészletek a következők:

- bos.net.ipsec.rte (A kernel IP biztonsági környezetének és parancsainak futási környezete)
- bos.msg.*LANG*.net.ipsec (ahol a *LANG* a megadott nyelv, például hu_HU)
- bos.net.ipsec.keymgt
- clic.rte (CryptoLite C, fájlkészlet DES, tripla DES és AES titkosítás esetén)

IKE digitális aláírás támogatásához telepítenie kell a gskit.rte fájlkészletet vagy a gskkm.rte fájlkészletet is a bővítőcsomagból.

Telepítése után az IP biztonság külön tölthető be az IPv4 és IPv6 protokollokhoz az “IP biztonság betöltése” helyen leírt ajánlott eljárással vagy az **mkdev** paranccsal.

IP biztonság betöltése:

SMIT használatával megvalósíthatja az IP biztonsági modulok automatikus betöltését az IP biztonság indításakor. A SMIT azt is biztosítja, hogy a kernel kiterjesztés és az IKE démonok a helyes sorrendben legyenek betöltve.

Megjegyzés: Az IP biztonság betöltése engedélyezi a szűrési funkciót. Betöltés előtt meg kell győződni róla, hogy a megfelelő szűrőszabályok létrehozása megtörtént. Ellenkező esetben elképzelhető, hogy minden külső kommunikáció letiltásra kerül.

Sikeres betöltés esetén az **lsdev** parancs az IP biztonsági eszközöket Elérhető állapotúnak tünteti fel.

```
lsdev -C -c ipsec

   ipsec_v4 Available IP Version 4 Security Extension
   ipsec_v6 Available IP Version 6 Security Extension
```

Az IP biztonság kernelbővítmény betöltése után az alagutak és szűrők készen állnak a beállításra.

IP biztonsági szolgáltatás konfigurációjának tervezése

Az IP biztonsági szolgáltatás beállításához először az alagutak és szűrők beállítását tervezze meg.

Ha az összes forgalom egy egyszerű alagutat használ, akkor a szűrőszabályok automatikusan is előállíthatók. Összetettebb szűrési igény esetén a szűrőszabályokat külön-külön be kell állítani.

Az IP biztonságot a Virtuális magánhálózat bedolgozóval vagy a Rendszergazdai kezelőfelülettel (SMIT) konfigurálhatja. SMIT használata esetén a következő gyorselérések használhatók:

smit ips4_basic
Alapszintű IPv4 beállítás

smit ips6_basic
Alapszintű IPv6 beállítás

Az IP biztonság beállításának megkezdése előtt meg kell határoznia a használandó módszereket, például hogy inkább alagutakat vagy szűrőket használna, vagy hogy melyik alagúttípus felelne meg legjobban az adott környezet igényeinek, stb. A döntések meghozatala előtt az alábbi szakaszokból szerezhet további információkat:

Hardveres gyorsítás:

A 10/100 Mbps Ethernet PCI Adapter II (termékkód: 4962) szabványokon alapuló IP biztonságot nyújt, hogy levegye az IP biztonsági funkciókkal kapcsolatos terheket az AIX operációs rendszerről.

Ha a 10/100 Mbps Ethernet PCI Adapter II megtalálható az AIX rendszerben, akkor az IP biztonsági verem kihasználja a kártya képességeit:

- Titkosítás és visszafejtés DES vagy tripla DES algoritmussal
- Hitelesítés az MD5 vagy SHA1 algoritmussal
- Biztonsági megegyezésekkel kapcsolatos információk tárolása.

A szoftveres algoritmus helyett a kártya funkciói kerülnek felhasználásra. A 10/100 Mbps Ethernet PCI Adapter II kézi és IKE alagutaknál is használható.

Az IP biztonság hardveres gyorsítási szolgáltatása a bos.net.ipsec.rte és devices.pci.1410ff01.rte fájlkészletek 5.1.0.25 vagy újabb változatánál érhető el.

A hálózati csatolóra áthelyezhető biztonsági megegyezések számára fogadás módban (bejövő forgalom) vonatkozik egy korlátozás. A küldési oldalon (kimenő forgalom) a támogatott konfigurációt használó összes csomagot a csatoló végzi. Bizonyos alagút konfigurációk nem helyezhetők át a csatolóra.

A 10/100 Mbps Ethernet PCI Adapter II a következő szolgáltatásokat támogatja:

- DES, 3DES vagy NULL titkosítás ESP esetén
- HMAC-MD5 vagy HMAC-SHA1 hitelesítés ESP vagy AH (de nem mindkettő) esetén (Ha az ESP és AH is használatban van, akkor az ESP-t kell először végrehajtani. IKE alagutaknál ez mindig így van, kézi alagutaknál azonban a felhasználó kiválaszthatja a sorrendet.)
- Szállítás és alagút mód
- IPv4 csomagok átvétele

Megjegyzés: A 10/100 Mbps Ethernet PCI Adapter II nem tudja kezelni az IP beállításokkal rendelkező csomagokat.

Ha engedélyezni kívánja a 10/100 Mbps Ethernet PCI Adapter II számára az IP biztonság feldolgozást, akkor elképzelhető, hogy le kell választani a hálózati csatolót, és engedélyezni kell az IP biztonság átvállalási szolgáltatást.

A hálózati csatoló leválasztásához a SMIT felület segítségével tegye a következőket:

Az IP biztonság átterhelés engedélyezéséhez tegye a következőket a SMIT felületen:

1. Jelentkezzen be **root** felhasználóként.
2. Írja be a parancssorba a smitty eadap parancsot, majd nyomja meg az Entert.
3. Válassza ki az **Ethernet kártya jellemzőinek módosítása / megjelenítése** menüpontot, majd nyomja meg az Entert.
4. Válassza ki a 10/100 Mbps Ethernet PCI Adapter II eszközt, majd nyomja meg az Entert.
5. Módosítsa az IP biztonság átterhelés mezőt az igen értékre, majd nyomja meg az Entert.

A hálózati csatoló parancssori leválasztásához írja be a következő parancsot:

```
# ifconfig enX detach
```

Az IPsec átterhelés attribútum parancssorból végzett engedélyezéséhez írja be a következő parancsot:

```
# chdev -l entX -a ipsec_offload=yes
```

Az IPsec átterhelés attribútum engedélyezésének ellenőrzéséhez a parancssorba írja be a következő parancsot:

```
# lsattr -El entX detach
```

Az IPsec átterhelési attribútum parancssori letiltásához írja be a következő parancsot:

```
# chdev -l entX -a ipsec_offload=no
```

Az **enstat** paranccsal ellenőrizheti, hogy az alagút konfiguráció használja-e az IP biztonság átterhelési szolgáltatást. Az **enstat** parancs az IP biztonság átterhelés engedélyezésekor megjeleníti a küldött és fogadott IP biztonsági csomagok összes statisztikáját. Ha például az Ethernet csatoló az **ent1**, akkor írja be a következő parancsot:

```
# entstat -d ent1
```

A kimenet a következőhöz hasonló lesz:

```
.
.
.
10/100 Mbps Ethernet PCI Adapter II (1410ff01) Specific Statistics:
--------------------------------------------
.
.
.
Transmit IPsec packets: 3
Transmit IPsec packets dropped: 0
Receive IPsec packets: 2
Receive IPsec packets dropped: 0
```

Hálózati hangolható paraméter:

A konfigurációjában lévő alagutak számától függően növelheti egy socket maximális pufferméretét.

Ha a környezetében nagyszámú alagút fut, és az **sb_max** hangolható paraméter az alapértelmezett értéken marad, akkor előfordulhat, hogy az IKE démon folyamat és a Tunnel Manager démon folyamat nem válaszol a hálózat nagy terhelése miatt.

Az **sb_max** hangolható paraméterhez a következő értékeket érdemes használni:

- 10 MB 500 alagút esetén
- 20 MB 1000 alagút esetén

Kapcsolódó tájékoztatás:

Az sb_max hangolható paraméter

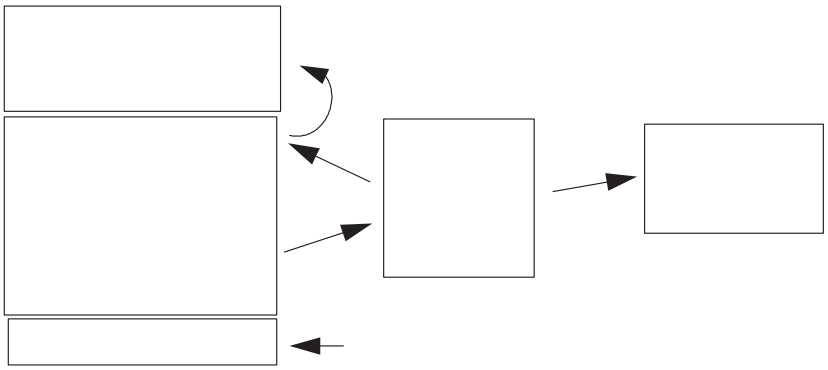
Alagutak és szűrők:

Az IP biztonsági szolgáltatás két különálló része az *alagutak* és a *szűrők*. Az alagutak igényelnek szűrőket, a szűrőkhöz azonban nincs szükség alagutakra.

A *szűrés* egy olyan funkció, amely lehetővé teszi, hogy a bejövő és kimenő csomagokat a rendszer *szabályoknak* nevezett különféle jellemzők alapján elutasítsa vagy elfogadja. A funkció segítségével a rendszeradminisztrátorok felügyelhetik a hoszt és más hosztok közötti forgalmat. A szűrés különféle csomagtulajdonságok, például forrás- és célcím, IP verziószám, alhálózati maszk, protokoll, port, útválasztási jellemzők, töredezettség, csatoló és alagút meghatározás alapján történhet. A szűrést az IP réteg végzi, így az alkalmazásokat nem kell módosítani.

Az *alagutak* két hoszt közötti biztonsági megegyezéseket határoznak meg. A biztonsági megegyezések különféle biztonsági paramétereket egyeztetnek az alagút két végpontján.

Az ábra bemutatja, hogyan érkeznek be a csomagok a hálózati csatolóról az IP veremhez. Itt meghívásra kerül a szűrő modul, amely meghatározza, hogy a csomag engedélyezett vagy tiltott. Ha van megadott alagút azonosító, akkor a rendszer összehasonlítja a csomagot a meglévő alagút meghatározásokkal. Ha az alagút kibontás sikeres, akkor a felsőbb szintű protokoll megkapja a csomagot. A funkció végrehajtása fordított sorrendben történik kimenő csomagok esetén. Az alagút funkció egy szűrőszabály segítségével társítja a csomagot az adott alagúthoz, ettől függetlenül szűrési tevékenység anélkül is folyhat, hogy a csomagokat egy alagút kapná.



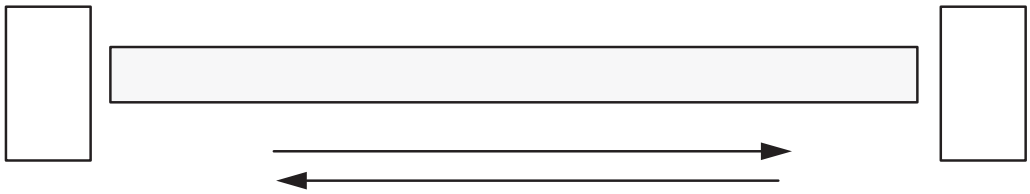
7. ábra: Hálózati csomagtovábbítás

Az ábra a hálózati csomagok által megtett útvonalat mutatja be. A hálózatból bejövő csomag a hálózati csatolón lép be. Innen bekerül az IP verembe, ahonnan a szűrő modulba kerül. A szűrő modulból a csomag vagy az alagút meghatározásokhoz kerül, vagy visszakerül az IP veremhez, amely továbbítja a felsőbb szintű protokollokhoz.

Alagutak és biztonsági megegyezések:

Alagutak kerülnek felhasználásra minden olyan esetben, amikor az adatok hitelesítése vagy titkosítása és hitelesítése szükséges. Az alagutak meghatározása két hoszt közötti biztonsági megegyezés megadásával történik. A biztonsági megegyezések határozzák meg a titkosítási és hitelesítési algoritmusokat, illetve az alagút jellemzőit.

Az alábbi illusztráció egy virtuális alagutat mutat be az A és B hoszt között.



8. ábra: Biztonságos alagút kialakítása az A és B hoszt között

Az ábra egy virtuális alagutat mutat be az A és B hoszt között. A B biztonsági megegyezés a B hosztból az A hosztba mutató nyíl. A biztonsági megegyezések egy célcímet, biztonsági paraméter indexet, kulcsot, titkosítási algoritmust és formátumot, hitelesítési algoritmust és kulcs élettartamot adnak meg.

A biztonsági megegyezéseket a biztonsági paraméter index (SPI) és a célcím határozza meg egyedi módon. Ezek a paraméterek szükségesek az alagutak egyedi azonosításához. A további paraméterek, úgymint a titkosítási algoritmus, hitelesítési algoritmus, a kulcsok és az élettartam meghatározható, vagy az alapértelmezések is használhatók.

Alagúttal kapcsolatos szempontok:

Számos dolgot figyelembe kell vennie, mielőtt eldöntené, hogy milyen alagúttípust használja az IP biztonsághoz.

Az IKE alagutak különböznek a kézi alagutaktól, mivel a biztonsági irányelvek konfigurálása és az alagút végpontjainak meghatározása külön folyamat.

Az IKE használatakor egy kétlépéses egyeztetési folyamat zajlik le. Az egyeztetési folyamat lépéseit *fázisnak* nevezzük, és mindkét fázis eltérő biztonsági stratégiákat alkalmaz.

Az Internet kulcscsere egyeztetés kezdetekor ki kell alakítani egy biztonságos csatornát az egyeztetésekhez. Ezt hívjuk *kulcskezelési* fázisnak vagy *1. fázisnak*. A fázis során mindkét fél egy előzetesen megosztott kulcs vagy digitális igazolás segítségével azonosítja a másikat, és adja át az azonosítási információkat. Ez a fázis egy olyan biztonsági megegyezést alakít ki, amelyben a felek meghatározzák a biztonságos kommunikációt, és megegyeznek a második fázisban alkalmazott kommunikáció védelmében. Ezen fázis eredménye egy *IKE*, más néven *1. fázis* alagút.

A második lépést *adatkezelési* fázisnak, vagy *2. fázisnak* nevezzük; ez az IKE alagút segítségével létrehozza a tényleges forgalom védelmére szolgáló AH és ESP biztonsági megegyezéseket. A második fázis határozza meg az IP biztonság alagutat használó adatokat is. A következők meghatározása történhet például:

- Alhálózati maszk
- Címtartomány
- Protokoll és portszám kombináció

9. ábra: IKE alagút beállítási folyamat

A következő ábra bemutatja az IKE alagutak kétlépéses, kétfázisos folyamatát.

Megjegyzés: Az IKEv2 szintén két fázissal rendelkezik. Az első fázis neve *IKE SA* fázis vagy *1. fázis*, a másodiké pedig *CHILD SA* fázis vagy *2. fázis*. Az alagutak IKEv1-ben való kialakításának módjával ellentétben az IKEv2-ben 1-es fázisú alagút kialakításakor automatikusan aktiválásra kerül egy 2-es fázisú alagút. Az IKEv2 alagutak konfigurációja az IKEv1 alagutak beállításához hasonló.

A kulcskezelési (IKE) alagút végpontjai sok esetben megegyeznek az adatkezelési (IP biztonság) alagút végpontjaival. Az IKE alagút végpontjai az egyeztetést végző számítógépek azonosítói. Az IP biztonság alagút végpontjai írják le az IP biztonság alagutat használó forgalom típusát. Egyszerű, hosztok közötti alagutaknál, amelyeknél a két alagút közti teljes forgalmat ugyanaz az alagút védi, az 1. és 2. alagút végpontok megegyeznek. Amikor az egyeztető felek átjárók, akkor az IKE alagút végpontjai az átjárók, az IP biztonság alagút végpontjai pedig az átjárók mögötti számítógépek, alhálózatok vagy alagút felhasználói címek.

Kulcskezelési paraméterek és stratégia:

A kulcskezelési stratégiát az IKE egyeztetés során használandó paraméterek megadásával szabhatja testre. Például külön kulcskezelési stratégiák adhatók meg megosztott kulcs és aláírás alapján végzett hitelesítéshez. Az 1. fázisnál a felhasználónak meg kell határoznia bizonyos kulcskezeléssel kapcsolatos biztonsági tulajdonságokat, amelyek alapján az adatcsere történik.

Az 1. fázis (kulcskezelési fázis) az IKE alagútkonfiguráció következő paramétereit határozza meg:

Kulcskezelési (1. fázis) alagút
Az IKE alagút neve. Minden egyes alagútnál meg kell határozni az egyeztetés végpontjait. Ez az a két gép, amely az IKE üzenetek küldését és ellenőrzését végzi. Az alagút neve meghatározhatja az alagút végpontokat, például VPN Budapest vagy VPN Szeged.

Hoszt azonosság típusa

Az IKE adatcsere során felhasznált azonosítótípus. A megfelelő kulcs kikeresés végrehajtásának biztosítása érdekében az azonosító típusának és értékének meg kell egyeznie az előzetesen megosztott kulcs értékeivel. Ha egy előzetesen megosztott kulcs érték kereséséhez önálló azonosító kerül felhasználásra, akkor a *hoszt azonosító* a kulcs azonosítója, *típusa* pedig KEY_ID. A KEY_ID típus akkor hasznos, ha egy hoszt egynél több előzetesen megosztott kulcs értékkel rendelkezik.

Hoszt azonosság

A hoszt azonosító típusának értéke IP címként, teljes képzésű tartománynévként vagy *felhasználó@tartománynév* címként kifejezve. Például: jdoe@studentmail.ut.edu.

IP cím

A távoli hoszt IP címe. Erre az értékre akkor van szükség, ha a hoszt azonosító típusa KEY_ID, vagy a hoszt azonosító típusa nem oldható fel IP címmé. Ha például a felhasználó neve nem oldható fel a helyi névszerveren, akkor a távoli oldalnak meg kell adni a IP címét.

Adatkezelési paraméterek és stratégia:

Az adatkezelési ajánlás paraméterei az IKE alagút meghatározásának 1. fázisában kerülnek beállításra. Ezek a kézi alagutaknál használtakkal megegyező IP biztonsági paraméterek írják le az alagút adatforgalmának védelmét. Ugyanazon 1. fázisú alagútban egynél több 2. fázisú alagút is indítható.

A következő végpont azonosító típusok írják le az IP biztonság alagutat használó adatok típusát:

Hoszt, Alhálózat vagy Tartomány
Megadja, hogy az alagúton haladó adatok célja egy adott hoszt, alhálózat vagy címtartomány.

Hoszt/alhálózat azonosító
Megadja az alagút felett forgalmat továbbító helyi és távoli rendszerek hoszt vagy alhálózat azonosságát. Meghatározza a 2. fázisú egyeztetés során küldött azonosítókat és a sikeres egyeztetés esetén kialakított szűrőszabályokat.

Alhálózati maszk
Leírja az alhálózat összes IP címét (például 9.53.250.96 hoszt és 255.255.255.0 maszk).

IP címtartomány kezdete
Megadja az alagutat használó címek tartományának kezdő IP címét.

IP címtartomány vége
Megadja az alagutat használó címek tartományának befejező IP címét.

Port Megadja, hogy az adatok egy adott portszámot (például 21 vagy 23) használnak.

Protokoll
Megadja, hogy az adatokat egy adott protokoll (például TCP vagy UDP) szállítja. Meghatározza a 2. fázisú egyeztetés során küldött protokollt és a sikeres egyeztetés esetén kialakított szűrőszabályokat. A helyi végpont és a távoli végpont protokolljának meg kell egyeznie egymással.

Befejező port
Leírja az adatátvitel befejező portját (például 100 vagy 500). Alapértelmezés szerint 65355 a befejező port.

Korlátozás: IKEv2 esetén csak IPv4 vagy IPv6 címtartományt használjon forgalomkiválasztóként. A befejező port csak IKEv2 és AIX 6.1 TL 04 vagy újabb esetén alkalmazható.

Alagúttípus kiválasztása:

A kézi vagy IKE alagutak használatával kapcsolatos döntés a távoli végpont alagút támogatásától, illetve az alkalmazni kívánt kulcskezelés típusától függ.

Ahol csak lehetséges, használjon IKE alagutakat, mivel ezek ipari szabványos biztonságos kulcsegyeztetést és -frissítést biztosítanak. Emellett kihasználják az ESP és AH fejléctípusok előnyeit, továbbá támogatják az újraküldés elleni védelmet is. Választhatóan beállíthatja az aláírás módot is, amellyel biztosíthatja digitális igazolások használatát.

Ha a távoli végpont kézi alagutat igénylő algoritmust használ, akkor kézi alagutakat kell használni. A kézi alagutak igen sokféle hoszttal használhatók együtt. Mivel a kulcsok statikusak, cseréjük pedig bonyolult és összetett folyamat, nem is annyira biztonságosak. Kézi alagutak jelen operációs rendszert futtató hosztok, és bármilyen más, IP biztonság támogatással rendelkező hosztok között kialakíthatók, feltéve, hogy van mindkét részről támogatott titkosítási és hitelesítési algoritmus. A legtöbb szállító megoldása kulcsolt MD5/DES vagy HMAC MD5/DES algoritmusokat biztosít. Ezek az IP biztonságnak szinte valamennyi megvalósításával működnek.

A kézi alagutak beállításakor alkalmazott eljárás attól függ, hogy az alagút első hosztját vagy második hosztját állítja be; ez utóbbinak ugyanis azonos paraméterekkel kell rendelkeznie. Az első hoszt beállításakor a kulcsok automatikusan is előállíthatók, és használhatók az alapértelmezett algoritmusok. A második hoszt beállításakor amikor csak lehet, importálja a távoli végpont alagút információit.

Másik fontos szempont annak meghatározása, hogy a távoli rendszer tűzfal mögött van-e. Ha igen, akkor beállításnak ki kell térnie a tűzfalra is.

IKE használata DHCP környezetben vagy dinamikusan hozzárendelt címekkel:

Az IP biztonság használatának egyik általános példája, amikor egy távoli rendszer IKE szekciót kezdeményez egy szerverrel, de azonosságuk nem köthető egy adott IP címhez.

Ez történhet helyi hálózati (LAN) környezetben is, például amikor egy számítógép IP biztonság protokollal csatlakozik egy LAN szerverre az adatok titkosításához. Egy másik gyakori eset, amikor egy kliens felhívja a szervert, és azonosításként egy teljes képzésű tartománynevet vagy egy e-mail címet (felhasználó@tartomány) ad meg.

A Kulcskezelés fázisban (első fázis) csak az RSA aláírás a támogatott hitelesítési mód, ha nem IP cím azonosítókkal használja a fő módot. Más szavakkal, ha előzetesen megosztott kulcs hitelesítést szeretne használni, akkor agresszív vagy fő módot kell használnia, és az IP címeket azonosítóként kell alkalmaznia. Ha nagyszámú DHCP klienssel szeretne IPsec alagutat létrehozni, akkor nem praktikus egyedi, előzetesen megosztott kulcsot definiálni minden egyes DHCP klienshez, így ebben a helyzetben ajánlott az RSA aláírás hitelesítés használata. Az alagút meghatározásban távoli azonosítóként használhatja a csoport azonosítót, így a csatornát csak egyszer kell definiálnia az összes DHCP klienssel (lásd: /usr/samples/ipsec/group_aix_responder.xml alagút definíciós példafájl). A csoport azonosító az AIX IPsec egyedi jellemzője. A csoport azonosítóba belefoglalhatja az IKE azonosítókat (például egy IP címet), az FQDN-t, a Felhasználói FQDN-t, egy IP cím tartományt vagy készletet és így tovább, majd a csoport azonosítót használhatja az első vagy második fázis azonosítójának az alagút meghatározásokban.

Megjegyzés: Ha csoport azonosítót használ, akkor a csatornát Csak válaszadóként kell meghatározni. Ez azt jelenti, hogy ezt az alagutat a DHCP kliens oldaláról kell aktiválnia.

Az adatkezelési fázis során amikor az IP biztonsági megegyezések TCP vagy UDP forgalmat titkosítanak, akkor beállítható egy általános adatkezelési alagút. Ennek megfelelően az 1. fázisban hitelesített valamennyi kérés az adatkezelési fázis számára beállított általános alagutat fogja használni, ha az IP cím nincs kifejezetten megadva az adatbázisban. Ez lehetővé teszi, hogy az általános alagútnak bármilyen cím megfeleljen, és használható legyen mindaddig, amíg az 1. fázis szigorú biztonsági ellenőrzései sikeresek.

Általános adatkezelési alagút megadása XML segítségével:

Az általános adatkezelési alagutakat az **ikedb** által használt XML formátumban lehet elvégezni.

Az **ikedb** felületről és az ikedb parancsról további információkat a “Parancssori felület IKE alagútkonfigurációhoz” oldalszám: 226 című szakaszban talál. Az általános adatkezelési alagutak DHCP környezetben kerülnek felhasználásra. Az XML formátum a az IPSecTunnel címkenevet használja. Más szövegkörnyezetekben ezt *2. fázisú alagútnak* is nevezik. Az *általános adatkezelés alagút* valójában nem alagút, hanem egy olyan IPSecProtection, amely akkor kerül

felhasználásra, amikor egy adott kulcskezelési alagúthoz tartozó bejövő adatkezelési üzenet nem felel meg a kulcskezelési alagútban meghatározott egyik adatkezelési alagútnak sem. Ez csak olyan esetben kerül felhasználásra, amikor az AIX rendszer a válaszadó. IPSecProtection általános adatkezelési alagút meghatározása nem kötelező.

Az általános adatkezelési alagutat az IKEProtection elem határozza meg. Erre két XML attribútum, az *IKE_IPSecDefaultProtectionRef* és *IKE_IPSecDefaultAllowedTypes* használható.

Először is meg kell határozni egy olyan IPSecProtection-t, amely alapértelmezésként használható abban az esetben, ha egyik IPSecTunnels (adatkezelési alagút) sem felel meg. Az alapértelmezésként használt IPSecProtection IPSec_ProtectionName attribútumának _defIPSprot_ karaktersorozattal kell kezdődnie.

Most az alapértelmezett IPSecProtection-t használó IKEProtection-t tekintjük. Adjon meg egy olyan **IKE_IPSecDefaultProtectionRef** attribútumot, amely tartalmazza az alapértelmezett IPSec_Protection nevét.

Jelen IKEProtection esetén meg kell adni az **IKE_IPSecDefaultAllowedTypes** attribútum értékét is. Ez az alábbi értékeket tartalmazhatja (több érték esetén ezeket szóközzel kell elválasztani egymástól):

```
Local_IPV4_Address
Local_IPV6_Address
Local_IPV4_Subnet
Local_IPV6_Subnet
Local_IPV4_Address_Range
Local_IPV6_Address_Range
Remote_IPV4_Address
Remote_IPV6_Address
Remote_IPV4_Subnet
Remote_IPV6_Subnet
Remote_IPV4_Address_Range
Remote_IPV6_Address_Range
```

Ezek az értékek a kezdeményező által megadott azonosító típusoknak felelnek meg. Az IKE egyeztetés során a tényleges azonosítók figyelmen kívül maradnak. A megadott **IKE_IPSecDefaultAllowedTypes** kerül felhasználásra, ha az IKE_IPSecDefaultAllowedTypes attribútum tartalmaz Local_ kezdetű karaktersorozatot, amely megfelel a kezdeményező helyi azonosító típusának és tartalmaz egy Remote_ kezdetű karaktersorozatot, amely megfelel a kezdeményező távoli azonosító típusának. Más szavakkal minden **IKE_IPSecDefaultAllowedTypes** attribútumban lennie kell legalább egy Local_ és Remote_ értéknek a megfelelő IPSec_Protection használatához.

Általános adatkezelési csatorna példa:

Adatkezelő alagút segítségével üzenetet küldhet a rendszernek.

Egy kezdeményező a következőt küldi az AIX rendszernek egy 2. fázisú (adatkezelés) üzenetben:

```
local ID type:    IPV4_Address
local ID:         192.168.100.104

remote ID type:   IPV4_Subnet
remote ID:        10.10.10.2
remote netmask:   255.255.255.192
```

Az AIX rendszer nem rendelkezik olyan adatkezelési alagúttal, amely megfelelne ezen azonosítóknak. Rendelkezik viszont egy olyan IPSecProtection alagúttal, amelyben a következő attribútumok vannak megadva:

```
IKE_IPSecDefaultProtectionRef="_defIPSprot_protection4"
IKE_IPSecDefaultAllowedTypes="Local_IPV4_Address
                              Remote_IPV4_Address
                              Remote_IPV4_Subnet
                              Remote_IPV4_Address_Range"
```

A bejövő üzenet helyi azonosító típusa IPV4_Address, amely megfelel a megengedett típusok egyik Local_ értékével, a Local_IPV4_Address bejegyzéssel. Emellett az üzenet távoli azonosítója (IPV4_Subnet) is megfelel a Remote_IPV4_Subnet bejegyzésnek. Ennek megfelelően az adatkezelési alagút egyeztetés folytatódik oly módon, hogy a _defIPSprot_protection4 lesz az IPSecProtection.

Az /usr/samples/ipsec/default_p2_policy.xml fájl egy teljes XML fájl, amely tartalmazza egy példaként használható általános IPSecProtection meghatározását.

Internet kulcscsere alagutak beállítása

Internet kulcscsere (IKE) alagutakat a Rendszergazdai kezelőfelületen (SMIT) vagy parancssorból konfigurálhat.

SMIT felület használata IKE alagút konfigurálására:

Az IKE alagutak meghatározására és az IKE adatbázissal kapcsolatos alapvető funkciók végrehajtására a SMIT felületen is lehetőség van.

A SMIT az XML parancsfunkciókat használja az IKE alagút meghatározások hozzáadására, törlésére és módosítására. Az IKE SMIT segítségével az IKE alagutak gyorsan beállíthatók, emellett példákat nyújt az IKE alagút meghatározások létrehozásához szükséges XML szintaxisról. Az IKE SMIT menük emellett lehetővé teszik az IKE adatbázis mentését, visszaállítását és inicializálását is.

IPv4 IKE alagút beállításához használja a **smitty ike4** gyorselérést. IPv6 IKE alagút beállításához használja a **smitty ike6** gyorselérést. Az IKE adatbázissal kapcsolatos funkciók az IP biztonság további beállításai menüben találhatók.

Parancssori felület IKE alagútkonfigurációhoz:

Az **ikedb** parancs segítségével a felhasználó egy XML felület segítségével lekérheti, frissítheti, törölheti, importálhatja és exportálhatja a IKE adatbázisban lévő információkat.

Az **ikedb** parancs lehetővé teszi a felhasználónak az IKE adatbázis írását (put) és olvasását (get). A bemeneti és kimeneti formátum egy Bővíthető leírónyelv (XML) fájl. Az XML fájlt formátumát annak Dokumentumtípus meghatározása (DTD) adja meg. Az **ikedb** parancs lehetővé teszi a felhasználónak az adatbázis írásához használt XML fájl ellenőrzését végző DTD megtekintését is. Bár a **-e** kapcsolóval lehetőség van egyed deklarációk hozzáadására a DTD fájlhoz, ez a DTD egyetlen módosítási lehetősége. Az XML fájl minden külső DOCTYPE deklarációja figyelmen kívül marad, és minden belső DOCTYPE deklaráció hibához vezethet. Az XML fájl DTD felhasználásával végzett elemzését előíró szabályokat az XML szabvány határozza meg. Az /usr/samples/ipsec példafájl egy tipikus XML fájl, amely meghatároz néhány általános alagút helyzetet. A szintaxis részletes leírását a *Commands Reference* **ikedb** paranccsal foglalkozó része tartalmazza.

Az **ike** parancs használható az IKE alagutak indítására, leállítására és megfigyelésére. Az **ike** parancs emellett használható az IKE és IP biztonsági alagutak aktiválására, eltávolítására és kilistázására is. A szintaxis részletes leírását a *Commands Reference* **ike** paranccsal foglalkozó része tartalmazza.

Az alábbi példák bemutatják az **ike**, **ikedb** és számos más parancs használatát egy IKE alagút beállítása és állapotának ellenőrzése során:

1. Az alagútegyeztetés megkezdéséhez (egy alagút *aktiválásához*) vagy (a megadott szereptől függően) a válaszadás engedélyezéséhez használja az **ike** parancsot az alagút számával az alábbiak szerint:

   ```
   # ike cmd=activate numlist=1
   ```

   A parancsban használhatja a távoli azonosítót vagy IP címet is, amint az a következő példákban is látható:

   ```
   # ike cmd=activate remid=9.3.97.256
   # ike cmd=activate ipaddr=9.3.97.100, 9.3.97.256
   ```

   Mivel a parancsok befejezése több percig is tarthat, a parancsok az egyeztetés megkezdése után visszatérnek.
2. Egy alagút állapotának megtekintéséhez használja az **ike** parancsot az alábbiak szerint:

   ```
   # ike cmd=list
   ```

   A kimenet a következőhöz hasonló lesz:

```
Phase 1 Tunnel ID       [1]
Phase 2 Tunnel ID       [1]
```

A kimenet azt mutatja, hogy az 1. és 2. fázisú alagutak jelenleg aktívak.

3. Az alagút állapotának részletes megjelenítéséhez az **ike** parancsot a következőképpen kell használni:

```
# ike cmd=list verbose
```

A kimenet a következőhöz hasonló lesz:

```
Phase 1 Tunnel ID       1
Local ID Type:          Fully_Qualified_Domain_Name
Local ID:               bee.austin.ibm.com
Remote ID Type:         Fully_Qualified_Domain_Name
Remote ID:              ipsec.austin.ibm.com
Mode:                   Aggressive
Security Policy:        BOTH_AGGR_3DES_MD5
Role:                   Initiator
Encryption Alg:         3DES-CBC
Auth Alg:               Preshared Key
Hash Alg:               MD5
Key Lifetime:           28800 Seconds
Key Lifesize:           0 Kbytes
Key Rem Lifetime:       28737 Seconds
Key Rem Lifesize:       0 Kbytes
Key Refresh Overlap:    5%
Tunnel Lifetime:        2592000 Seconds
Tunnel Lifesize:        0 Kbytes
Tun Rem Lifetime:       2591937 Seconds
Status:                 Active

Phase 2 Tunnel ID       1
Local ID Type:          IPv4_Address
Local ID:               10.10.10.1
Local Subnet Mask:      N/A
Local Port:             any
Local Protocol:         all
Remote ID Type:         IPv4_Address
Remote ID:              10.10.10.4
Remote Subnet Mask:     N/A
Remote Port:            any
Remote Portocol:        all
Mode:                   Oakley_quick
Security Policy:        ESP_3DES_MD5_SHA_TUNNEL_NO_PFS
Role:                   Initiator
Encryption Alg:         ESP_3DES
AH Transform:           N/A
Auth Alg:               HMAC-MD5
PFS:                    No
SA Lifetime:            600 Seconds
SA Lifesize:            0 Kbytes
SA Rem Lifetime:        562 Seconds
SA Rem Lifesize:        0 Kbytes
Key Refresh Overlap:    15%
Tunnel Lifetime:        2592000 Seconds
Tunnel Lifesize:        0 Kbytes
Tun Rem Lifetime:       2591962 Seconds
Assoc P1 Tunnel:        0
Encap Mode:             ESP_tunnel
Status:                 Active
```

4. Az újonnan aktivált alagút dinamikus szűrőtáblájában lévő szűrőszabályok megjelenítéséhez az **lsfilt** parancs használható az alábbiak szerint:

```
# lsfilt -d
```

A kimenet a következő példához hasonlóan néz ki:

```
1 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no udp eq 4001 eq 4001 both both no all
  packets 0 all
2 *** Dynamic filter placement rule *** no
0 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 yes all any 0 any 0 both both no all
  packets 0 all

*** Dynamic table ***

0 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no udp eq 500 eq 500 local both no all
  packets 0
0 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no ah any 0 any 0 both inbound no all
  packets 0
0 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no esp any 0 any 0 both inbound no all
  packets 0
1 permit 10.10.10.1 255.255.255.255 10.10.10.4 255.255.255.255 no all any 0 any
  0 both outbound yes all packets 1
1 permit 10.10.10.4 255.255.255.255 10.10.10.1 255.255.255.255 no all any 0 any
  0 both inbound yes all packets 1
```

A példa egy olyan számítógépre vonatkozik, amelyen egyetlen IKE alagút van, más alagút pedig nincs. A dinamikus szűrőelhelyezési szabályt (a fenti példa 2. szabálya a statikus táblában) a felhasználó áthelyezheti, ha az elhelyezést más felhasználó által megadott szabályok miatt módosítani szeretné. A dinamikus tábla szabályainak előállítása automatikusan történik az alagutak egyeztetése során. Ezek a szabályok megjeleníthetők ugyan, de nem módosíthatók.

5. A dinamikus szűrőszabályok naplózásának bekapcsolásához állítsa be a 2. szabály naplózási beállítását a **chfilt** paranccsal, az alábbi példában látható módon:

   ```
   # chfilt -v 4 -n 2 -l y
   ```

   Az IKE forgalom naplózásáról további részleteket a “Naplózási szolgáltatások” oldalszám: 251 szakaszban talál.
6. Az alagút leállításához használja az **ike** parancsot a következőképpen:

   ```
   # ike cmd=remove numlist=1
   ```
7. Az alagút meghatározások megjelenítéséhez használja az **ikedb** parancsot az alábbiak szerint:

   ```
   # ikedb -g
   ```
8. Ha az IKE adatbázis meghatározásait felül kívánja írni egy másik számítógépen előállított XML fájl tartalmával, akkor használja az **ikedb** parancsot az alábbiak szerint:

   ```
   # ikedb -pFs peer_tunnel_conf.xml
   ```

   A peer_tunnel_conf.xml a másik számítógépen előállított XML fájl.
9. Ha meg kívánja szerezni a *tunnel_sys1-sys2* nevű 1. fázisú alagút meghatározását az összes függő 2. fázisú alagúttal és a megfelelő ajánlásokkal és átalakításokkal együtt, akkor használja az **ikedb** parancsot az alábbi formában:

   ```
   # ikedb -gr -t IKETunnel -n tunnel_sys1_and_sys2
   ```
10. Az adatbázis összes előzetesen megosztott kulcsának törléséhez írja be a következő **ikedb** parancsot:

    ```
    # ikedb -d -t IKEPresharedKey
    ```

Az IKE alagút csoport támogatására vonatkozó általános leírást az alábbi rész tartalmaz: “Csoporttámogatás” oldalszám: 229. Az **ikedb** parancsot csoportok parancssori meghatározására is használhatja.

AIX IKE és Linux kapcsolat:

A Linux konfigurációs fájlok segítségével be lehet állítani egy AIX IKE alagutat.

Ha a Linux konfigurációs fájlok segítségével be kíván állítani egy AIX IKE alagutat, akkor az **ikedb** parancsot a **-c** kapcsolóval használja (átalakítás), amely lehetővé teszi a /etc/ipsec.conf és /etc/ipsec.secrets Linux konfigurációs fájl használatát IKE alagút meghatározásként. Az **ikedb** parancs elemzi a Linux konfigurációs fájlokat, létrehoz egy XML fájlt, és igény esetén hozzáadja az XML alagút meghatározásokat az IKE adatbázishoz. Az alagút meghatározások az **ikedb -g** paranccsal jeleníthetők meg.

Csoporttámogatás:

Az IP biztonság támogatja az alagút meghatározások IKE azonosítóinak csoportosítását, így több azonosító társítható egyetlen biztonsági irányelvhez és nem szükséges önálló alagútmeghatározásokat létrehozni.

A csoportosítás különösen hasznos számos távoli hoszthoz vezető kapcsolat beállításakor, mivel ilyenkor elkerülhető a sok különálló alagút meghatározás beállítása és kezelése. Emellett ha módosítani kell a biztonsági stratégián, akkor nincs szükség számos alagút meghatározás módosítására.

A csoportokat meg kell határozni, mielőtt nevük beállítható lenne az alagút meghatározásokban. A csoportok mérete 1 KB-ban korlátozott. Az egyeztetés kezdeményezői oldalán csak adatkezelési alagút meghatározásokban használhat csoportokat távoli azonosítóként. Az egyeztetés válaszadó oldalán kulcskezelési és adatkezelési alagút meghatározásokban is használhatók csoportok távoli azonosítóként.

A csoportok egy csoportnévből, illetve IKE azonosítókat és azonosítótípusokat tartalmazó listából állnak. Az azonosítók a következők lehetnek:

- IPv4 címek
- IPv6 címek
- Teljes képzésű tartománynevek
- Felhasználó@tartomány
- X500 DN típusok

A biztonsági megegyezés egyeztetés során a rendszer a csoportban szereplő azonosítókat sorban keresi az első egyezésig.

A csoportok parancssorból történő létrehozásával kapcsolatos információkat az alábbi rész tartalmaz: “Parancssori felület IKE alagútkonfigurációhoz” oldalszám: 226.

IKE alagút beállítási példahelyzetek:

Az alábbi példahelyzetek mutatják be az alagutak beállítását igénylő legáltalánosabb példahelyzeteket. A példahelyzetek jelentős része besorolható a telephely, üzleti partner és távoli hozzáférés kategóriákba.

- A telephely példahelyzet esetében adott két megbízható hálózat, például egy cég két különálló telephelyének hálózata, amelyet össze kell kapcsolni. Ebben a példában a hálózatokat átjárók kötik össze, és az átjárók közötti teljes forgalom ugyanazt az alagutat használja. Az alagút végén a forgalom kibontásra kerül, és szokásos formában bekerül a vállalati intranetbe.

  Az IKE egyeztetés első fázisában létrejön az IKE biztonsági megegyezés a két átjáró között. Az IP biztonsági alagúton áthaladó forgalom a két alhálózat közötti forgalom, és a 2. egyeztetési fázisban az alhálózat azonosítók kerülnek felhasználásra. A biztonsági stratégia és az alagút paramétereinek beírása után az alagút kap egy számot. Az alagút az **ike** paranccsal indítható el.
- Az üzleti partner példahelyzetnél a hálózatok nem megbízhatóak, ezért a hálózati adminisztrátor valószínűleg kis számú hosztra szeretné korlátozni a biztonsági átjáró mögötti hozzáférést. Ebben az esetben a hosztok közötti alagút IP biztonsággal védett forgalmat továbbít. A 2. fázisú alagút protokollja AH vagy ESP. Ezt a hoszt-hoszt alagutat egy átjáró-átjáró alagút védi.
- A távoli hozzáférés példahelyzetében az alagutak kialakítása igény szerint történik, magas biztonsági szint alkalmazásával. Ebben az esetben IP címek alkalmazása nem szerencsés, helyettük inkább a teljes képzésű tartománynevek vagy a *felhasználó@tartomány* azonosítók használhatók. Választhatóan egy KEYID segítségével a kulcsok hozzárendelhetők egy hoszt azonosítóhoz.

Digitális igazolások és kulcskezelési alapelvek

A digitális igazolások egy azonosságot és egy nyilvános kulcsot kötnek össze, és lehetővé teszik egy titkosított átvitel küldőjének vagy címzettjének ellenőrzését.

Az IP biztonság digitális igazolásokat használ a *nyilvános kulcsú kriptográfia* (más néven *aszimmetrikus kriptográfia*) támogatásához. Ennek lényege, hogy az adatok titkosítása a csak a felhasználó által ismert magánkulccsal történik, visszafejtésük pedig az adott nyilvános/magánkulcspár megfelelő nyilvános (megosztott) kulcsával. A *kulcspárok* hosszú adatsorozatok, amelyek egy felhasználó titkosítási sémáját jelentik.

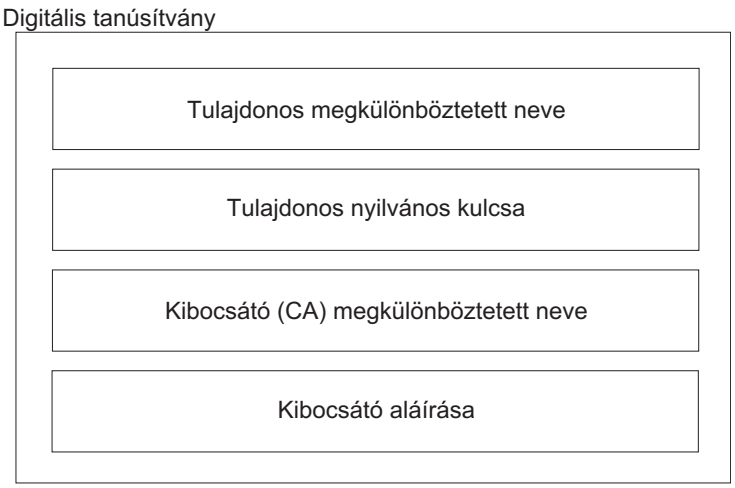
A nyilvános kulcsú kriptográfiában a nyilvános kulcs bárki számára hozzáférhető, akivel a felhasználó kommunikálni szeretne. A küldő digitálisan aláír minden biztonságos kommunikációt a kulcspár megfelelő magánkulcsával. A fogadó a nyilvános kulcs segítségével ellenőrzi a küldő aláírását. Ha az üzenet sikeresen visszafejthető a nyilvános kulccsal, akkor a fogadó ellenőrizheti a küldő hitelesítését.

A nyilvános kulcsú titkosítás a digitális igazolások kibocsátását megbízható külső felekre, *igazolási hatóságokra* (*CA*) bízza. A fogadó meghatározhatja, hogy mely kibocsátó szervezetek vagy hatóságok tekinthetők megbízhatónak. Az igazolások kiadása egy adott időszakra szól, az érvényesség lejárta után az igazolást ki kell cserélni.

Az AIX a Kulcskezelési eszközt biztosítja digitális tanúsítványok kezeléséhez. A következő szakaszok az igazolásokkal kapcsolatos alapfogalmakat mutatják be.

Digitális igazolások formátuma:

A digitális igazolás különféle információkat tartalmaz az igazolás tulajdonosának azonosságáról és az igazolási hatóságról. A digitális igazolásokat az alábbi ábra mutatja be.



A digitális igazolások tartalma

10. ábra: A digitális igazolások tartalma

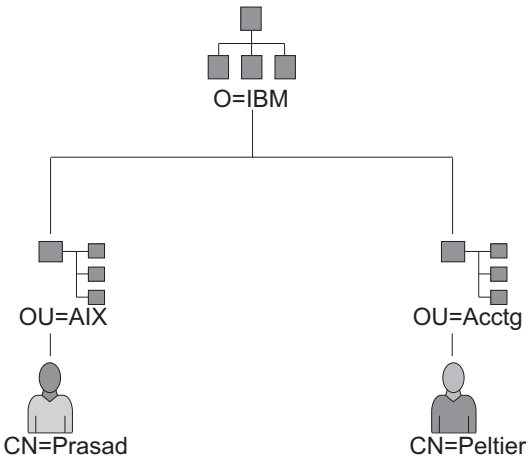
Ez az ábra mutatja be a digitális igazolások négy főbb elemét. Felülről lefelé ezek a következők: Tulajdonos megkülönböztetett neve, tulajdonos nyilvános kulcsa, kibocsátó megkülönböztetett neve és a kibocsátó aláírása.

A digitális igazolások tartalmát a következő szakasz tárgyalja részletesen:

Tulajdonos megkülönböztetett neve

Ez a tulajdonos általános nevének és címtárfában elfoglalt környezetének (helyének) kombinációja. Az alábbi egyszerű címtárfában például Prasad a tulajdonos általános neve, a környezet pedig ország=US, szervezet=ABC, szervezeti egység=SERV; ennek megfelelően a megkülönböztetett név:

```
/C=US/O=ABC/OU=SERV/CN=prasad.austin.ibm.com
```



Példa megkülönböztetett név címtárfából levezetésére

11. ábra: Példa megkülönböztetett név címtárfából levezetésére

Ez az illusztráció egy olyan címtárfát ábrázol, amelynek legfelső szintje az O=ABC, és két egyed benne a második szinten. A második szinten az OU=AIX és OU=Acctg ágak találhatók, ezek mindegyike egy további ágat tartalmaz a legalsó szinten. A legalsó szinten a CN=Prasad és a CN=Peltier található.

Tulajdonos nyilvános kulcsa
Ezt használják a címzettek az adatok visszafejtésére.

Tulajdonos alternatív neve
Ez lehet egy azonosító, például IP cím, e-mail cím, teljes képzésű tartománynév, stb.

Kiadás dátuma
A digitális igazolás kiadásának dátuma.

Lejárat dátuma
A digitális igazolás lejártának dátuma.

Kibocsátó megkülönböztetett neve
Az igazolási hatóság megkülönböztetett neve.

Kibocsátó digitális aláírása
Az igazolás érvényesítésére használható digitális aláírás.

Biztonsági szempontok digitális aláírásokhoz:

A digitális igazolások önmagukban nem bizonyítják az azonosságot.

A digitális igazolás csak a digitális igazolás azonosságának ellenőrzését teszi lehetővé a tulajdonos digitális aláírásának ellenőrzéséhez használható nyilvános kulcs megadásával. Nyilvános kulcsát nyugodtan elküldheti bárki másnak, mivel adatait nem lehet a kulcspár másik része, a magánkulcs nélkül visszafejteni. Ebből következően a tulajdonosnak meg kell védenie az igazolás nyilvános kulcsához tartozó magánkulcsot. A magánkulcs ismertté válása esetén a digitális igazolás tulajdonosának minden kommunikációja visszafejthető. A magánkulcs nélkül a digitális igazolások nem használhatók fel rossz szándékkal.

Igazolási hatóságok és bizalmi hierarchiák:

A digitális igazolások csak annyira megbízhatók, mint az azokat kibocsátó igazolási hatóságok.

A bizalom részeként meg kell ismerni az igazolások kibocsátásakor alkalmazott irányelveket. Minden szervezetnek és felhasználónak saját magának kell eldöntenie, hogy mely igazolási hatóságokat tekinti megbízhatónak.

A Kulcskezelési eszköz lehetővé teszi a szervezeteknek saját aláírású igazolások létrehozását, amely hasznos lehet kis számú felhasználóval vagy számítógéppel végzett tesztelés esetén.

Biztonsági szolgáltatás felhasználójaként ismernie kell annak magánkulcsát a digitális igazolások igényléséhez és érvényesítéséhez. Emellett egy digitális igazolás fogadása nem ellenőrzi annak hitelességét. A hitelesség ellenőrzéséhez szükség van a digitális igazolást kibocsátó igazolási hatóság nyilvános kulcsára. Ha még nem rendelkezik az igazolási hatóság nyilvános kulcsának megbízható másolatával, akkor elképzelhető, hogy egy további digitális igazolásra lesz szükség a hatóság nyilvános kulcsának megszerzéséhez.

Igazolás visszavonási listák:

A digitális igazolásokról feltételezik, hogy a teljes érvényességi időszakban használják azokat. Ha azonban szükség van rá, akkor az igazolások a tényleges lejárati dátum előtt is érvényteleníthetők.

Az igazolás érvénytelenítésére például az alkalmazott kilépése vagy az igazolás magánkulcsának ismertté válása esetén lehet szükség. Az igazolások érvénytelenítéséhez tudatni kell az igazolási hatósággal ennek körülményeit. Amikor a hatóság visszavon egy igazolást, akkor hozzáadja az érvénytelen igazolás sorozatszámát egy Igazolás visszavonási listához (CRL).

Az Igazolás visszavonási listák olyan aláírt adatszerkezetek, amelyeket az igazolási hatóságok rendszeres időközönként közzétesznek egy nyilvános helyen. A Igazolás visszavonási listák HTTP vagy LDAP szerverekről szerezhetők be. Minden egyes CRL rendelkezik egy aktuális időbélyeggel és egy következő frissítés időbélyeggel. A lista visszavont igazolásait az igazolások sorozatszáma azonosítja.

Ha IKE alagutat állít be digitális igazolásokon alapuló hitelesítésre, akkor az igazolás visszavonásának ellenőrzéséhez válassza ki az RSA aláírás CRL ellenőrzéssel beállítást. Ha a CRL ellenőrzés engedélyezett, akkor az egyeztetési folyamat részeként a rendszer lekérdezi a visszavont igazolások listáját is.

> **Megjegyzés:** A szolgáltatás használatához a rendszert be kell állítani egy SOCKS szerver (HTTP szerverek esetén 4. változat), egy LDAP szerver vagy mindkettő használatára. Ha tudja, hogy melyik SOCKS vagy LDAP szervert használja are using to obtain Igazolás visszavonási listák (CRL) beszerzéséhez, akkor hozzáadhatja azokat az /etc/isakmpd.conf fájlhoz

Digitális igazolások használata internetes alkalmazásokban:

A nyilvános kulcsú titkosítást alkalmazó internetes alkalmazásoknak digitális igazolásokat kell használniuk a nyilvános kulcsok megszerzéséhez.

Több alkalmazás is használ nyilvános kulcsú kriptográfiát, egyebek között a következők is:

Virtuális magánhálózatok (VPN)
A virtuális magánhálózatok, más néven *biztonságos alagutak* állíthatók be két rendszer, például tűzfal között, hogy biztonságos kommunikációt valósíthassanak meg nem biztonságos kommunikációs vonalakon. A végpontok közötti teljes hálózat titkosított formában történik.

Az alagútkezelés protokolljai az IP biztonság és IKE szabványokat használják, amelyek lehetővé teszik egy távoli kliens (például egy otthonról dolgozó alkalmazott) és egy biztonságos hoszt vagy hálózat közötti titkosított kommunikációt.

Védett socket réteg (SSL)
Az SSL protokoll kommunikációs bizalmasságot és integritást biztosít. Ezt használják egyebek között webszerverek a szerverek és böngészők közötti biztonságos kapcsolatokhoz, az Egyszerűsített címtárhozzáférési protokoll (LDAP) szerverek a kliensek kapcsolatainak titkosításához, és a Host-on-Demand a kliens és a hosztrendszer közötti kapcsolatokhoz. Az SSL digitális igazolásokat használ a kulcscseréhez, a szerver hitelestéshez és választhatóan a kliens hitelesítéshez.

Biztonságos elektronikus levelezés

Több PEM vagy S/MIME szabványt támogató elektronikus levelezési rendszer is használ digitális igazolásokat a digitális aláírások illetve az üzenetek titkosításához és visszafejtéséhez használt kulcsok cserjéhez.

Digitális igazolások és igazoláskérések:

Digitális igazolás igényléséhez létre kell hozni egy *igazolási kérést*, és el kell küldeni azt egy igazolási hatósághoz.

Az aláírt digitális igazolások tartalmazzák a tulajdonos megkülönböztetett nevének és nyilvános kulcsának, illetve az igazolási hatóság megkülönböztetett nevének és aláírásának mezőit. A saját aláírású digitális igazolások a tulajdonos megkülönböztetett nevét, nyilvános kulcsát és aláírását tartalmazzák.

Az igazolási kérés a kérelmező megkülönböztetett nevét, nyilvános kulcsát és aláírását tartalmazza. Az igazolási hatóság a digitális igazolás nyilvános kulcsával ellenőrzi a kérelmező aláírását, hogy megbizonyosodjon a következők felől:

- Az igazolási kérés nem változott meg a kérelmező és a CA közötti átvitel során.
- A kérelmező birtokában van az igazolási kérésben szereplő nyilvános kulcshoz tartozó magánkulcs.

Az igazolási hatóság emellett felelős a kérelmező azonosságának bizonyos szintű ellenőrzéséért is. Az ellenőrzés követelményei széles skálán mozoghatnak a tulajdonos azonosságának valamilyen gyenge bizonyítéka és az abszolút meggyőződés között.

Kulcskezelési eszköz:

A bővítőcsomag gskkm.rte fájlkészletében található Kulcskezelési eszköz kezeli a digitális igazolásokat.

A digitális igazolások és aláírások támogatásához legalább az 1., 2., 3,. 4., 6. és 7. feladatot el kell végezni. Azután hozza létre az IKE alagutat, és társítson egy irányelvet az alagúttal, mely RSA aláírást használ hitelesítési módszerként.

Létrehozhat és konfigurálhat kulcsadatbázist a certmgr paranccsal, a parancssorból a Kulcskezelési eszközt megnyitva.

Ez a szakasz a Kulcskezelés használatát írja le a következő feladatok végrehajtására:

Kulcsadatbázis létrehozása:

A kulcsadatbázis lehetővé teszi, hogy a VPN végpontok érvényes digitális igazolás felhasználásával csatlakozzanak egymáshoz. Az IP biztonság virtuális magánhálózatainál a kulcsadatbázis (.kdb) formátum kerül felhasználásra.

A Kulcskezelési eszköz az alábbi igazolási hatóságok digitális igazolásait tartalmazza:

- RSA Secure Server Certification Authority
- Thawte Personal Premium Certification Authority
- Thawte Personal Freemail Certification Authority
- Thawte Personal Basic Certification Authority
- Thawte Personal Server Certification Authority
- Thawte Server Certification Authority
- Verisign Class 1 Public Primary Certification Authority
- Verisign Class 2 Public Primary Certification Authority
- Verisign Class 3 Public Primary Certification Authority
- Verisign Class 4 Public Primary Certification Authority

Ezek az aláírt digitális igazolások teszik lehetővé a klienseknek, hogy csatlakozzanak olyan szerverekhez, amelyek ezen kibocsátók által kiadott érvényes digitális igazolásokkal rendelkeznek. A kulcsadatbázist a létrehozás után felhasználhatja olyan szerverre csatlakozáshoz, amelynek digitális igazolását a fenti aláírók valamelyike bocsátotta ki.

A listában nem szereplő aláíró digitális igazolás használatához be kell szereznie azt a hatóságtól, és hozzá kell adnia a kulcsadatbázishoz. Lásd: “Igazolási hatóság gyökér digitális igazolásának hozzáadása”.

Kulcsadatbázisnak a **certmgr** paranccsal végzett létrehozásához tegye a következőket:

1. Indítsa el a Kulcskezelési eszközt a következő parancs beírásával:
   ```
   # certmgr
   ```
2. A Kulcsadatbázis fájllistában válassza ki az **Új** lehetőséget.
3. A **Kulcsadatbázis típusa** mezőhöz fogadja el az alapértelmezett CMS kulcsadatbázis fájl értéket.
4. A **Fájlnév** mezőben adja meg a következő fájlnevet:
   ```
   ikekey.kdb
   ```
5. A **Hely** mezőben adja meg az adatbázis helyét:
   ```
   /etc/security
   ```

   Megjegyzés: A kulcsadatbázisnak az ikekey.kbd nevet kell kapnia és a /etc/security könyvtárba kell helyezni. Ellenkező esetben az IP biztonság nem működik megfelelően.
6. Kattintson az **OK** gombra. Megjelenik a **Jelszó** ablak.
7. Adjon meg egy jelszót a **Jelszó** mezőben, majd adja meg ismét a **Jelszó megerősítése** mezőben.
8. Ha módosítani kívánja a jelszó érvényességének alapértelmezett időtartamát, akkor adja meg a napok számát a **Lejárati idő beállítása?**mezőben. A mező alapértelmezett értéke 60 nap. Ha nem kívánja, hogy a jelszó lejárjon, akkor hagyja üresen a **Lejárati idő beállítása?** mezőt.
9. A jelszó titkosított változatának fájlba mentéséhez válassza ki a **Jelszó tárolása fájlban?** mezőt és írja be az Igen értéket.

   Megjegyzés: Ha a digitális igazolásokat az IP biztonsággal is használni kívánja, akkor jelszót tárolni kell.
10. Kattintson az **OK** gombra. Megjelenik egy megerősítés képernyő, amely hírül adja a kulcsadatbázis létrehozását.
11. Kattintson újra az **OK** gombra; ezzel visszakerül az IBM Kulcskezelés képernyőre. Innentől kezdve végrehajthat más feladatokat, vagy kiléphet az eszközből.

Igazolási hatóság gyökér digitális igazolásának hozzáadása:

Miután kérte és megkapta egy igazolási hatóság gyökér igazolását, hozzáadhatja azt az adatbázishoz.

A legtöbb gyökér digitális igazolás .arm kiterjesztéssel rendelkezik, például:
```
cert.arm
```

Az igazolási hatóság gyökér igazolásának adatbázishoz adásához tegye a következőket:

1. Ha a Kulcskezelés még nem fut, akkor indítsa el az eszközt a következő parancs beírásával:
   ```
   # certmgr
   ```
2. A főképernyőn válassza a Kulcsadatbázis fájl lista **Megnyitás** menüpontját.
3. Jelölje ki a használni kívánt kulcsadatbázis fájlt, majd kattintson a **Megnyitás** gombra.
4. Írja be a jelszót, majd kattintson az **OK** gombra. Ha a jelszót a program elfogadta, akkor visszatér az IBM Kulcskezelés képernyőre. A címsorban megjelenik a kijelölt kulcsadatbázis fájl neve, jelezve, hogy a fájl meg van nyitva, és lehetőség van annak használatára.
5. Válassza ki a **Személyes/aláíró igazolások** lista **Aláíró igazolások** menüpontját.
6. Kattintson a **Hozzáadás** gombra.
7. Az **Adattípus** listában válasszon ki egy adattípust, például:
   ```
   Base64 kódolású ASCII adatok
   ```

8. Adja meg az igazolási hatóság gyökér igazolásának helyét és fájlnevét, vagy kattintson a **Tallózás** gombra a fájl kiválasztásához.
9. Kattintson az **OK** gombra.
10. Adja meg az igazolási hatóság gyökér igazolásának címkéjét, például Teszt CA gyökér igazolás, majd kattintson az **OK** gombra. Visszakerül a **Kulcskezelés** képernyőre. Az **Aláíró igazolások** mezőben megjelenik a hozzáadott igazolási hatóság gyökér igazolása. Innentől kezdve végrehajthat további feladatokat, vagy kiléphet az eszközből.

Megbízhatósági beállítások kialakítása:

A telepített igazolási hatóság igazolások alapértelmezésben megbízható megjelölést kapnak. Ha szükséges akkor módosíthatja a megbízhatósági beállítást.

A megbízhatósági beállítás módosításához tegye a következőket:

1. Ha a Kulcskezelés még nem fut, akkor a következő parancs beírásával indítsa el az eszközt:
   ```
   # certmgr
   ```
2. A főképernyőn válassza a **Kulcsadatbázis fájl lista Megnyitás** menüpontját.
3. Jelölje ki a használni kívánt kulcsadatbázis fájlt, majd kattintson a **Megnyitás** gombra.
4. Írja be a jelszót, majd kattintson az **OK** gombra. A jelszót elfogadása után visszakerül az **IBM Kulcskezelés** képernyőre. A címsorban megjelenik a kijelölt kulcsadatbázis fájl neve, jelezve, hogy a fájl meg van nyitva.
5. Válassza ki a **Személyes/aláíró igazolások** lista **Aláíró igazolások** menüpontját.
6. Jelölje ki a módosítani kívánt igazolást, majd kattintson duplán a bejegyzésre, vagy kattintson a **Megjelenítés/szerkesztés** gombra. A **Kulcsinformációk** képernyő megjelenítésre kerül az igazolásbejegyzéshez.
7. Az igazolás megbízható gyökérigazolásként beállításához jelölje meg az **Igazolás beállítása megbízható gyökérként** mellett levő jelölőnégyzetet, majd kattintson az **OK** gombra. Ha az igazolás nem megbízható, akkor törölje a jelölőnégyzet kijelölését, és kattintson az **OK** gombra.
8. Kattintson az **Aláíró igazolások** képernyő **OK** gombjára. Visszakerül az **IBM kulcskezelés** képernyőre. Innentől kezdve végrehajthat más feladatokat, vagy kiléphet az eszközből.

Igazolási hatóság gyökér digitális igazolás törlése:

Ha az aláíró digitális igazolások lista valamelyik igazolási hatóságát a jövőben nem kívánja használni, akkor törölje a hatóság gyökér digitális igazolását.

Megjegyzés: Igazolási hatóság gyökér digitális igazolásának törlése előtt készítsen róla biztonsági másolatot, hátha esetleg a jövőben mégis szükség lenne rá.

Az igazolási hatóság gyökér igazolásának törléséhez tegye a következőket:

1. Ha a Kulcskezelés még nem fut, akkor a következő parancs beírásával indítsa el az eszközt:
   ```
   # certmgr
   ```
2. A főképernyőn válassza a **Kulcsadatbázis fájl** lista **Megnyitás** menüpontját.
3. Jelölje ki a használni kívánt kulcsadatbázis fájlt, majd kattintson a **Megnyitás** gombra.
4. Írja be a jelszót, majd kattintson az **OK** gombra. A jelszó elfogadása után visszatér a **Kulcskezelés** képernyőre. A címsorban megjelenik a kijelölt kulcsadatbázis fájl neve, jelezve, hogy a fájl meg van nyitva, és lehetőség van annak használatára.
5. Válassza ki a **Személyes/aláíró igazolások** lista **Aláíró igazolások** menüpontját.
6. Jelölje ki a törölni kívánt igazolást, majd kattintson a **Törlés** gombra. Megjelenik a **Megerősítés** képernyő.
7. Kattintson az **Igen** gombra. Visszakerül az **IBM kulcskezelés** képernyőre. Az igazolási hatóság gyökér igazolásának címkéje a továbbiakban nem jelenik meg az **Aláíró igazolások** mezőben. Innentől kezdve végrehajthat más feladatokat, vagy kiléphet az eszközből.

Digitális aláírás kérése:

Digitális igazolás beszerzéséhez hozzon létre egy kérést a Kulcskezelési eszközben, majd küldje el a kérést az igazolási hatósághoz. A létrejött kérési fájl PKCS#10 formátumban lesz. Az igazolási hatóság ellenőrzi az azonosságát, majd visszaküld egy digitális igazolást.

Digitális igazolás igényléséhez tegye a következőket:

1. Ha a Kulcskezelés még nem fut, akkor a következő parancs beírásával indítsa el az eszközt:
   ```
   # certmgr
   ```
2. A főképernyőn válassza a **Kulcsadatbázis fájl** lista **Megnyitás** menüpontját.
3. Jelölje ki az /etc/security/ikekey.kdb kulcsadatbázis fájlt, amelyből elő kívánja állítani a kérést, majd kattintson a **Megnyitás** gombra.
4. Írja be a jelszót, majd kattintson az **OK** gombra. A jelszót elfogadása után visszakerül az **IBM Kulcskezelés** képernyőre. A címsorban megjelenik a kijelölt kulcsadatbázis fájl neve, jelezve, hogy a fájl meg van nyitva, és lehetőség van annak használatára.
5. Válassza ki a **Létrehozás > Új igazoláskérés** lehetőséget.
6. Kattintson az **Új** gombra.
7. A következő képernyő adja meg a saját aláírású igazolás Kulcscímkéjét, például:
   ```
   keytest
   ```
8. Adjon meg egy általános nevet (az alapértelmezett a hosztnév) és szervezetet, majd válasszon egy országot. A többi mezőben meghagyhatja az alapértelmezett értéket is, de módosíthatja is azokat.
9. Adja meg a Tulajdonos alternatív nevét. A Tulajdonos alternatív mezővel társított nem kötelező mezők az e-mail cím, az IP cím és a DNS név. IP cím alapján kialakított alagutak esetén írja be ugyanazt az IP címet, mint az IKE alagút IP cím mezőjébe. *Felhasználó@tartomány* típusú alagút esetén töltse ki az e-mail cím mezőt. Teljes képzésű tartománynév azonosítótípusú alagutak esetén írja be a teljes képzésű tartománynevet (például *hosztnev.vallalatnev*.hu) a DNS név mezőbe.
10. A képernyő alján adjon meg egy nevet a fájlnak, például:
    ```
    certreq.arm
    ```
11. Kattintson az **OK** gombra. Megjelenik egy megerősítés képernyő, amely hírül adja az új digitális igazolási kérés létrehozását.
12. Kattintson az **OK** gombra. Visszakerül az **IBM kulcskezelés** képernyőre. A **Személyes igazolási kérések** mezőben megjelenik a létrehozott digitális igazolási kérés (PKCS#10) kulcscímkéje.
13. Digitális igazolás igényléséhez küldje el a fájlt az igazolási hatóságnak. Innentől kezdve végrehajthat más feladatokat, vagy kiléphet az eszközből.

Új digitális igazolás hozzáadása (fogadása):

Miután megkapta a hatóságtól az új digitális igazolást, hozzá kell adnia azt ahhoz a kulcsadatbázishoz, amelyből a kérést előállította.

Új digitális igazolás hozzáadásához (fogadásához) tegye a következőket:

1. Ha a Kulcskezelés még nem fut, akkor indítsa el az eszközt a következő parancs beírásával:
   ```
   # certmgr
   ```
2. A főképernyőn válassza a **Kulcsadatbázis fájl** lista **Megnyitás** menüpontját.
3. Válassza ki az igazolási kérés előállításához használt kulcsadatbázist, majd kattintson a **Megnyitás** gombra.
4. Írja be a jelszót, majd kattintson az **OK** gombra. A jelszót elfogadása után visszakerül az IBM Kulcskezelés képernyőre. A címsorban megjelenik a kijelölt kulcsadatbázis fájl neve, jelezve, hogy a fájl meg van nyitva, és lehetőség van annak használatára.
5. Válassza ki a **Személyes/aláíró igazolások** lista **Személyes igazolás kérések** menüpontját.
6. Az újonnan kapott digitális igazolás adatbázishoz adásához kattintson a **Fogadás** gombra.

7. Az **Adattípus** listában válassza ki az új digitális igazolás adattípusát. Az alapértelmezés a **Base64 kódolású ASCII adatok**.
8. Adja meg az új digitális igazolás helyét és fájlnevét, vagy kattintson a **Tallózás** gombra a fájl kiválasztásához.
9. Kattintson az **OK** gombra.
10. Adja meg az új digitális igazolás leíró címkéjét, például:

    ```
    VPN telephely igazolás
    ```

11. Kattintson az **OK** gombra. Visszakerül az **IBM kulcskezelés** képernyőre. A **Személyes igazolások** mezőben megjelenik a hozzáadott új digitális igazolás. Innentől kezdve végrehajthat más feladatokat, vagy kiléphet az eszközből. Ha az igazolás betöltése során hiba történik, akkor győződjön meg róla, hogy az igazolást tartalmazó fájl a ———BEGIN CERTIFICATE——— szöveggel kezdődik, és az ———END CERTIFICATE——— szöveggel végződik.

    Például:

    ```
    -----BEGIN CERTIFICATE-----
    ajdkfjaldfwwwwwwwwwwadafdw
    kajf;kdsajkflasasfkjafdaff
    akdjf;ldasjkf;safdfdasfdas
    kaj;fdljk98dafdas43adfadfa
    -----END CERTIFICATE-----
    ```

    Ha a szöveg nem így néz ki, akkor módosítsa az igazolás fájlt, hogy a megfelelő szöveggel kezdődjön és végződjön.

Digitális igazolás törlése:

Néha törölni kell a digitális igazolást.

Megjegyzés: Digitális igazolások törlése előtt készítsen róluk biztonsági másolatot, hátha mégis szükség lesz rájuk a jövőben.

Digitális igazolások adatbázisból törléséhez tegye a következőket:

1. Ha a Kulcskezelés még nem fut, akkor a következő parancs beírásával indítsa el az eszközt:

    ```
    # certmgr
    ```

2. A főképernyőn válassza a **Kulcsadatbázis fájl** lista **Megnyitás** menüpontját.
3. Jelölje ki a használni kívánt kulcsadatbázis fájlt, majd kattintson a **Megnyitás** gombra.
4. Írja be a jelszót, majd kattintson az **OK** gombra. A jelszót elfogadása után visszakerül az **IBM Kulcskezelés** képernyőre. A címsorban megjelenik a kijelölt kulcsadatbázis fájl neve, jelezve, hogy a fájl meg van nyitva, és lehetőség van annak használatára.
5. Válassza ki a **Személyes/aláíró igazolások** lista **Személyes igazolás kérések** menüpontját.
6. Jelölje ki a törölni kívánt digitális igazolást, majd kattintson a **Törlés** gombra. Megjelenik a **Megerősítés** képernyő.
7. Kattintson az **Igen** gombra. Visszakerül az **IBM kulcskezelés** képernyőre. A törölt digitális igazolás címkéje a továbbiakban nem jelenik meg a **Személyes igazolások** mezőben. Innentől kezdve végrehajthat más feladatokat, vagy kiléphet az eszközből.

Adatbázisjelszó módosítása:

Néha meg kell változtatni az adatbázisjelszót.

A kulcsadatbázis módosításához tegye a következőket:

1. Ha a Kulcskezelés még nem fut, akkor a következő parancs beírásával indítsa el az eszközt:

    ```
    # certmgr
    ```

2. A főképernyőn válassza a **Kulcsadatbázis fájl** lista **Jelszómódosítás** menüpontját.
3. Adja meg az új jelszót a **Jelszó** mezőben, majd írja be ismét a **Jelszó megerősítése** mezőben.

4. Ha módosítani kívánja a jelszó érvényességének alapértelmezett időtartamát, akkor adja meg a napok számát a **Lejárati idő beállítása?**mezőben. A mező alapértelmezett értéke 60 nap. Ha nem kívánja, hogy a jelszó lejárjon, akkor ürítse ki a **Lejárati idő beállítása?** mezőt.
5. A jelszó titkosított változatának fájlba mentéséhez válassza ki a **Jelszó tárolása fájlban?** mezőt és adja meg az Igen értéket.

   Megjegyzés: Ha a digitális igazolásokat az IP biztonsággal is használni kívánja, akkor jelszót tárolni kell.
6. Kattintson az **OK** gombra. Az állapotsorban megjelenő üzenet tudatja a kérés sikeres befejezését.
7. Kattintson újra az **OK** gombra; ezzel visszakerül az **IBM Kulcskezelés** képernyőre. Innentől kezdve végrehajthat más feladatokat, vagy kiléphet az eszközből.

IKE alagutak létrehozása digitális igazolásokkal:

Digitális igazolásokat használó IKE alagút létrehozásához RSA aláírást kell megadnia hitelesítési módként az IKE alagút átalakítási irányelv-fájlban.

A következő példa RSA aláírásokat meghatározó XML irányelv-fájlt mutat be:

```
<!-- irányelv meghatározása az IKE alagút számára -->
<IKEProtection
 IKE ProtectionName="ike_3des_sha">
 <IKETTransform
  IKE AuthenticationMethod="RSA_signatures"
  IKE Encryption="3DES-CBC"
  IKE Hash="SHA"
  IKE DHGroup="1"/>
</IKEProtection>
```

Az IP biztonság az alábbi IKE alagút hoszt azonosság típusokat támogatja:

- IP cím
- teljes képzésű tartománynév (FQDN)
- *felhasználó@tartomány*
- X.500 megkülönböztetett név
- Kulcsazonosító

Ha az IKE alagút RSA aláírás módot használ, akkor jellemzően az X.500 megkülönböztetett nevek kerülnek felhasználásra az IKE alagút meghatározásban. Például ha az alagút helyi és a távoli hosztja a **/C=US/O=ABC/OU=SERV/CN=localname.austin.ibm.com** és a**/C=US/O=ABC/OU=SERV/CN=remotename.austin.ibm.com**, akkor az IKE alagút meghatározás az XML fájlban a következőhöz hasonló:

```
<IKETunnel>
 IKE TunnelName="Key_Tunnel"
 IKE ProtectionRef="ike_3des_sha">
<IKELocalIdentity>
 <ASN1_DN Value="/C=US/O=ABC/OU=SERV/CN=localname.austin.ibm.com">
 </ASN1_DN>
</IKELocalIdentity>
<IKERemoteIdentity>
 <ASN1_DN Value="/C=US/O=ABC/OU=SERV/CN=remotename.austin.ibm.com">
 </ASN1_DN>
</IKERemoteIdentity>
</IKETunnel>
```

A szükséges igazolásnak a hitelesítő szervezettől (CA) való beszerzéséhez használja a Kulcskezelési eszközt az igazoláskérés előállítására. Például az igazolásban a tulajdonos megkülönböztetett neveként **/C=US/O=ABC/OU=SERV/CN=name.austin.ibm.com** használata esetén az alábbi értékeket kell megadnia a kulcskezelési eszközben digitális igazoláskérés létrehozása során:

Általános név
name.austin.ibm.com

Szervezet
ABC

Szervezeti egység
SERV

Ország
US

A megadott X.500 megkülönböztetett név az a név, ami jellemzően beállít a rendszer- vagy LDAP adminisztrátor. A szervezeti egység értéke nem kötelező.

Az IP biztonság a digitális igazolásban támogatja tulajdonos alternatív neveként más azonosságtípusok megadását is. Például ha az alternatív hoszt azonosságként a 10.10.10.1 IP címet használja, akkor az alábbi értékeket kell megadnia a digitális igazoláskérésben:

Általános név
name.austin.ibm.com

Szervezet
ABC

Szervezeti egység
SERV

Ország
US

Tulajdonos alternatív IP címe mező
10.10.10.1

Miután létrehozta a fenti információkkal rendelkező igazolási kérést, a CA ezeket az információkat használja fel a személyes digitális igazolás létrehozásához.

Személyes digitális igazolás igénylésekor az igazolási hatóságnak a következő információkra van szüksége:

- Egy X.509 igazolást kér.
- Az aláírás formátuma MD5, RSA titkosítással.
- Megad-e alternatív nevet. Az alternatív név típusait a következő lista tartalmazza:
  - IP cím
  - teljes képzésű tartománynév (FQDN)
  - *felhasználó@tartomány*

  A tulajdonos alternatív név információi bekerülnek az igazolási kérés fájlba.
- A kulcs tervezett használata (a digitális aláírás bitet ki kell választani).
- A Kulcskezelő digitális igazolási kérés fájlja (PKCS#10 formátumban).

A kulcskezelési eszköz igazoláskérés létrehozásához való felhasználásával kapcsolatos információkért lásd: “Digitális aláírás kérése” oldalszám: 236.

Mielőtt aktiválná az IKE alagutat, az igazolási hatóságtól kapott személyes digitális igazolást hozzá kell adnia az ikekey.kdb kulcsadatbázishoz. További információk: “Új digitális igazolás hozzáadása (fogadása)” oldalszám: 236.

Az IP biztonság a személyes digitális igazolások következő típusait támogatja:

Tulajdonos megkülönböztetett neve
A tulajdonos megkülönböztetett nevét a következő formátumban és sorrendben kell megadni:

/C=US/O=ABC/OU=SERV/CN=*név*.austin.ibm.com

A Kulcskezelési eszköz csak egy **OU** értéket engedélyez.

Tulajdonos megkülönböztetett név és alternatív név IP címként
A tulajdonos megkülönböztetett neve és alternatív neve megjelölhető IP címként, amint azt a következő példa is bemutatja:

/C=US/O=ABC/OU=SERV/CN=név.austin.ibm.com és 10.10.10.1

Tulajdonos megkülönböztetett név és alternatív név teljes képzésű tartománynévként
A tulajdonos megkülönböztetett neve és alternatív neve megjelölhető teljes képzésű tartománynévként, amint azt a következő példa is bemutatja:

/C=US/O=ABC/OU=SERV/CN=név.austin.ibm.com és bell.austin.ibm.com.

Tulajdonos megkülönböztetett név és alternatív név *felhasználó@tartomány* **formában**
A tulajdonos megkülönböztetett neve és alternatív neve megjelölhető felhasználói címként (*felhasználó@tartománynév*), amint azt a következő példa is bemutatja:

/C=US/O=ABC/OU=SERV/CN=név.austin.ibm.com és név@austin.ibm.com.

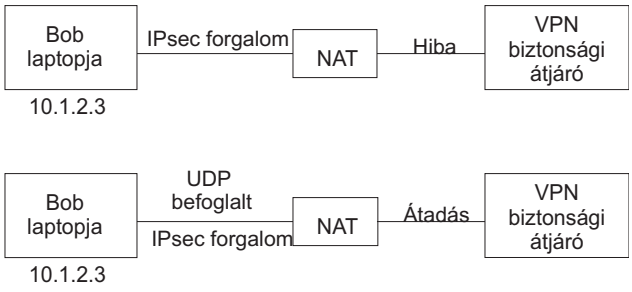
Tulajdonos megkülönböztetett név és több alternatív név
A tulajdonos megkülönböztetett neve több alternatív névvel is társítható, ezt a következő példa mutatja be:

/C=US/O=ABC/OU=SERV/CN=név.austin.ibm.com és bell.austin.ibm.com, 10.10.10.1 és user@név.austin.ibm.com.

Hálózati cím fordítás

Az IP biztonsági szolgáltatás azokat az eszközöket használhatja, amelyek címe hálózati cím fordításon (NAT) mennek keresztül.

A hálózati cím fordítást széles körben használják a tűzfal technológiában az internet kapcsolatok megosztásához, és szabványos szolgáltatás az útválasztókon és az egyéb külső eszközökön. Az IP biztonság protokoll a távoli végpontok azonosításától és azok távoli IP címén alapuló házirendjének meghatározásától függ. Ha közbenső eszközök - útvonalkezelők, tűzfalak - fordítják a saját címet nyilvános címre, akkor az IP biztonság szükséges hitelesítés feldolgozása meghiúsulhat, mivel az IP csomagban található cím a hitelesítési kivonat kiszámítása után módosításra került. Az új IP biztonság NAT támogatással a hálózati címfordítást végző csomópontok mögé beállított eszközök képesek IP biztonsági csatornát létrehozni. Az IP biztonság kódja észleli, ha egy távoli cím fordításra került. Az új IP biztonság megvalósítás és a NAT támogatás együttes használata lehetővé teszi a VPN kliensek számára, hogy otthonról vagy bárhonnan csatlakozzanak az irodához egy engedélyezett NAT szolgáltatással rendelkező internet kapcsolaton keresztül.



12. ábra: NAT szolgáltatással rendelkező IP biztonság

Az ábra a NAT szolgáltatással rendelkező, UDP beágyazott forgalmat használó IP biztonság megvalósítás és a NAT nélküli megvalósítás közötti különbséget mutatja be.

IP biztonsági szolgáltatás beállítása hálózati cím fordítással:

Ha NAT támogatást szeretne használni az IP biztonsági szolgáltatásban, akkor az /etc/isakmpd.conf fájlban be kell állítania az *ENABLE_IPSEC_NAT_TRAVERSAL* változót. Ha ez a változó be van állítva, akkor a rendszer szűrőszabályokat ad hozzá a forgalom 4500-as porton keresztüli küldéséhez és fogadásához.

Az alábbi példa az *ENABLE_IPSEC_NAT_TRAVERSAL* változó beállításakor használt szűrőszabályokat mutatja.

```
Dynamic rule 2:
Rule action          : permit
Source Address      : 0.0.0.0 (any)
Source Mask         : 0.0.0.0 (any)
Destination Address : 0.0.0.0 (any)
Destination Mask    : 0.0.0.0 (any)
Source Routing      : no
Protocol             : udp
Source Port         : 0 (any)
Destination Port    : 4500
Scope               : local
Direction            : inbound
Fragment control     : all packets
Tunnel ID number     : 0

Dynamic rule 3:
Rule action          : permit
Source Address      : 0.0.0.0 (any)
Source Mask         : 0.0.0.0 (any)
Destination Address : 0.0.0.0 (any)
Destination Mask    : 0.0.0.0 (any)
Source Routing      : no
Protocol             : udp
Source Port         : 4500
Destination Port    : 0 (any)
Scope               : local
Direction            : outbound
Fragment control     : all packets
Tunnel ID number     : 0
```

Az *ENABLE_IPSEC_NAT_TRAVERSAL* változó beállítása a szűrőtáblához is további szűrőszabályokat ad hozzá. A speciális IPSEC NAT üzenetek UDP beágyazást használnak, és az ilyen forgalom engedélyezéséhez szűrőszabályokat kell hozzáadni. Ezenkívül az első fázisban aláírás módra van szükség. Ha az igazolásban IP címet használ azonosítóként, akkor az igazolásnak a saját IP címet kell tartalmaznia.

Az IP biztonság NAT kapcsolatfenntartási üzeneteket küld, így tartja fennt a leképezést az eredeti IP cím és a NAT cím között. Az időtartamot a /etc/isakmpd.conf fájl *NAT_KEEPALIVE_INTERVAL* változója adja meg. Ez a változó határozza meg másodpercekben, hogy a rendszer milyen gyakran küld NAT kapcsolatfenntartási csomagokat. Ha nem ad meg értéket a *NAT_KEEPALIVE_INTERVAL* paraméterben, akkor a rendszer az alapértelmezett 20 másodperces beállítást használja.

Korlátozások NAT csere használatakor:

A NAT eszközök mögötti végpontoknak ESP protokollal kell védeniük a saját forgalmukat.

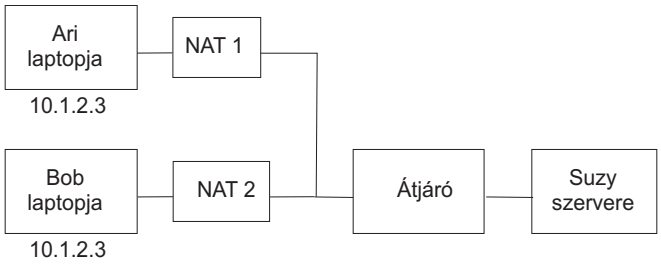
Az ESP az IP biztonság fő fejléce, és a legtöbb alkalmazás képes használni. Az ESP titkosított felhasználói adatokat tartalmaz, de az IP fejlécet nem. Az AH fejléc integritás ellenőrzése magában foglalja az IP forrás és cél címet a kulcsolt üzenet integritás ellenőrzésben. A cím mezőket módosító NAT vagy fordított NAT eszközök érvénytelenítik az üzenet integritási ellenőrzést. Ezért ha csak az AH protokoll van definiálva a csatorna második fázisú házirendjében és a NAT-ot a rendszer az első fázisú cserekor észleli, akkor egy NO_PROPOSAL_CHOSEN hasznos tartalom értesítés kerül elküldésre.

Ezenkívül a NAT-ot használó kapcsolatnak a alagút módot kell választania, hogy az eredeti IP cím be legyen ágyazva a csomagba. Az átviteli mód és a címek a NAT-tal nem kompatibilisek. Ha az átvitel NAT-ot észlel és csak a második fázisú átviteli mód kerül ajánlásra, akkor egy NO_PROPOSAL_CHOSEN hasznos tartalom értesítés kerül elküldésre.

Alagút mód konfliktusok elkerülése:

A távoli felek olyan bejegyzéseket egyeztethetnek, amelyek átfedésre kerülnek egy átjárón. Az átfedés az alagút mód ütközéséhez vezet.

Az alábbi ábra egy alagút mód ütközést ábrázol.



13. ábra: Alagút mód ütközés

Az átjárónak két lehetséges Biztonsági társítása (SA) van a 10.1.2.3 IP címhez. A két távoli cím zavart okoz, mert a rendszer nem tudja, hogy hova kell küldeni a szerverről érkező csomagokat. A Suzy szervere és Ari laptopja közötti alagút beállításakor az IP címet kell használni, és Suzy nem állíthat be alagutat Bobbal ugyanazzal az IP címmel. A csatorna mód ütközések elkerülése érdekében ne definiáljon alagutat azonos IP címmel. Mivel a távoli cím nem a távoli felhasználó irányítása alatt van, ezért egyéb azonosító típust kell használni a távoli hoszt azonosításához, például egy teljes képzésű tartománynevet vagy egy teljes képzésű tartománynév felhasználóját.

Kézi alagutak beállítása

Beállíthatja az IP biztonság kézi alagutakat, ha az eszközök nem támogatják az automatikus beírási módszert.

Kézi alagutak és szűrők:

Az alagút beállításának menete úgy néz ki, hogy először be kell állítani az alagutat az egyik végponton, azután importálni kell a beállításokat a másik végponton, majd aktiválni kell az alagutat és a szűrőszabályokat mindkét végponton. Az alagút ezzel készen áll a használatra.

Kézi alagutak beállításához nincs feltétlenül szükség a szűrőszabályok kézi meghatározására. Ha a két hoszt közötti teljes forgalom az alagúton halad át, akkor a szükséges szűrőszabályok automatikusan létrejönnek.

Az alagútra vonatkozó információknak mindkét oldalon meg kell egyezniük, ha nincsenek kifejezetten megadva. Például a forrásnál megadott titkosítási és hitelesítési algoritmusok kerülnek felhasználásra a célban is, ha a célra vonatkozó értékek nincsenek meghatározva.

Kézi alagút létrehozása az első hoszton:

Alagutat konfigurálhat a SMITips4_basic gyorseléréssel (IP v4 esetén), a SMIT ips6_basic gyorseléréssel (IP v6 esetén), vagy létrehozhatja saját kezűleg az alagutat a következőkkel.

A következő példában egy kézi alagutat hozunk létre a **gentun** paranccsal:

```
gentun -v 4 -t manual -s 5.5.5.19 -d 5.5.5.8 \
   -a HMAC_MD5 -e DES_CBC_8 -N 23567
```

Az előző példában létrehozott kézi alagút jellemzőinek listázására az **lstun -v 4** parancs használható. A kimenet a következő példához hasonlóan néz ki:

```
Tunnel ID            : 1
IP Version           : IP Version 4
Source               : 5.5.5.19
Destination          : 5.5.5.8
Policy               : auth/encr
Tunnel Mode          : Tunnel
Send AH Algo         : HMAC_MD5
Send ESP Algo        : DES_CBC_8
Receive AH Algo      : HMAC_MD5
Receive ESP Algo     : DES_CBC_8
Source AH SPI        : 300
Source ESP SPI       : 300
Dest AH SPI          : 23576
Dest ESP SPI         : 23576
Tunnel Life Time     : 480
Status               : Inactive
Target               : -
Target Mask          : -
Replay               : No
New Header           : Yes
Snd ENC-MAC Algo     : -
Rcv ENC-MAC Algo     : -
```

Az alagút aktiválásához írja be a következő kódot:

```
mktun -v 4 -t1
```

Az alagúthoz tartozó szűrőszabályok automatikusan létrejönnek.

A szűrőszabályok megjelenítésére használja az **lsfilt -v 4** parancsot. A kimenet a következő példához hasonlóan néz ki:

```
Rule 4:
Rule action          : permit
Source Address        : 5.5.5.19
Source Mask           : 255.255.255.255
Destination Address   : 5.5.5.8
Destination Mask      : 255.255.255.255
Source Routing       : yes
Protocol              : all
Source Port           : any 0
Destination Port      : any 0
Scope                : both
Direction             : outbound
Logging control      : no
Fragment control     : all packets
Tunnel ID number      : 1
Interface            : all
Auto-Generated        : yes

Rule 5:
Rule action          : permit
Source Address        : 5.5.5.8
Source Mask           : 255.255.255.255
Destination Address   : 5.5.5.19
Destination Mask      : 255.255.255.255
Source Routing       : yes
Protocol              : all
Source Port           : any 0
Destination Port      : any 0
Scope                : both
Direction             : inbound
Logging control      : no
```

```
Fragment control     : all packets
Tunnel ID number      : 1
Interface            : all
Auto-Generated        : yes
```

A szűrőszabályok aktiválásához - az alapértelmezett szabályokat is beleértve - használja az **mktun -v 4 -t 1** parancsot.

A másik oldal beállításához lehetőség van az alagút meghatározásának exportálására, majd importálására a másik számítógépen, amennyiben az is ezt az operációs rendszert használja.

A következő parancs exportálja az alagútdefiníciót a **-f** kapcsolóval jelzett könyvtár **ipsec_tun_manu.exp** fájljába, a hozzá tartozó szűrőszabályokat pedig az **ipsec_fltr_rule.exp** fájlba:

```
exptun -v 4 -t 1 -f /tmp
```

Kézi alagút létrehozása a második hoszton:

Az alagút megfelelő végpontjának létrehozásához az exportált fájlok átmásolásra és importálásra kerülnek a távoli számítógépre:

A következő parancs segítségével hozza létre az alagút megfelelő végpontját:

```
imptun -v 4 -t 1 -f /tmp
```

ahol

1 Az importálni kívánt alagút

/tmp Az importálandó fájlokat tartalmazó könyvtár

Az alagút számát a rendszer állítja elő. A számot a **gentun** parancs kimenetéből tudhatja meg, vagy listázza ki az alagutakat az **lstun** paranccsal, és állapítsa meg a megfelelő importálandó alagút számát. Ha az importálandó fájlban csak egy alagút található, vagy az összes alagutat importálni szeretné, akkor a **-t** kapcsoló nem szükséges.

Ha a távoli számítógép is ezt az operációs rendszert használja, akkor az export fájl használható az algoritmus, a kulcs és a biztonsági paraméter index (SPI) értékeknek az alagút másik végének megfelelő beállítására.

A tűzfal termékek export fájljai importálhatók alagút létrehozási céllal. Ehhez használja a **-n** kapcsolót a fájl importálásakor:

```
imptun -v 4 -f /tmp -n
```

Szűrők eltávolítása:

A szűrők eltávolításához és az IP biztonság leállításához használja az **rmdev** parancsot.

Az alapértelmezett szűrőszabály még akkor is aktív marad, ha a szűrést az **mkfilt -d** paranccsal kikapcsolja. Ez a parancs lehetővé teszi az összes szűrőszabály felfüggesztését vagy eltávolítását és új szabályok betöltését, miközben az alapértelmezett szabály védelme megmarad. Az alapértemezett szűrőszabály a *DENY*. Ha leállítja a szűrést az **mkfilt -d** paranccsal, akkor az **lsfilt** parancsból származó jelentések azt fogják jelezni hogy a szűrés ki van kapcsolva, de a csomagokat a rendszer nem engedi sem ki sem be. Ha teljesen ki szeretné kapcsolni az IP biztonságot, akkor használja az **rmdev** parancsot.

IP biztonsági szűrő beállítása

A szűrés beállítható egyszerűen, szinte kizárólag automatikusan előállított szűrőszabályokkal, de lehetőség van arra is, hogy rendkívül pontos szűrési funkciókat érjünk el az IP csomagok különféle tulajdonságai alapján végzett szűréssel.

A szűrőtábla sorait *szabályoknak* nevezzük. A szabályok gyűjteménye határozza meg, hogy mely csomagok fogadhatók el, és hová kell ezeket irányítani. A bejövő csomagok szűrőszabályainak megfeleltetését a rendszer a forráscímnek és az SPI értéknek a szűrőtáblában felsorolt értékekkel való összehasonlításával végzi. Ennek megfelelően az említett

párosításnak egyedinek kell lennie. A szűrőszabályok a kommunikáció több szempontját is felügyelhetik, így a forrás- és célcímeket és maszkokat, a protokollt, a portszámot, az irányt, a töredezettséget, a forrás útválasztást, az alagutat valamint a csatoló típusát.

A szűrőszabályok típusai a következők lehetnek:

- A Statikus szűrőszabályok jellemzően általános forgalomszűrésre, vagy a forgalomnak kézi alagutakhoz társítására szolgálnak. Ezek kiegészíthetők, törölhetők, módosíthatók és áthelyezhetők. A szabályok azonosítását segítendő minden szabály kiegészíthető egy szöveges leírással is.
- Az Automatikusan előállított és felhasználói szűrőszabályok (más néven *automatikusan előállított* szűrőszabályok) IKE alagutakhoz létrehozott adott szabálykészletek. A statikus és dinamikus szűrőszabályok is az adatkezelési alagút információin és az adatkezelés alagút egyeztetésén alapulnak.
- Az Előre meghatározott szűrőszabályok olyan általános szűrőszabályok, amelyek nem módosíthatók, helyezhetők át vagy törölhetők; ilyen például a teljes forgalom szabály, az ah szabály és az esp szabály. Ezek a teljes forgalomra vonatkoznak.

A **genfilt** parancs irány kapcsolóját (**-w**) kell megadni ha adott szabályt kell alkalmazni bemenő csomag feldolgozáskor vagy kimenő csomag feldolgozáskor. A kapcsoló **mindkét** értékének használata azt jelzi, hogy a szabályt a bemeneti és a kimeneti feldolgozáskor is használni kell. Ha a szűrés be van kapcsolva az AIX IPsec-ben, akkor legalább egy szabály határozza meg a hálózati csomagok sorsát (bejövő és kimenő csomagoknál is). Ha csak egy bejövő (vagy kimenő) csomag feldolgozásakor szeretne használni egy szabályt, akkor ezt megadhatja a **genfilt** parancs **-w** kapcsolójával. Ha például egy csomagot küld A hosztról B hosztra, akkor a kimenő IP csomag forráscíme az *A*, célcíme a *B* hoszt lesz. Ezt a csomagot az A hoszton az IPsec szűrő a kimenő feldolgozásban dolgozza fel, a B hoszton pedig a bejövő feldolgozásban. Tegyük fel hogy az A és B hoszt között található a G átjáró. A G átjárón ez a csomag (minden állandó meg értéke azonos) kétszer kerül feldolgozásra: egyszer a bejövő feldolgozásban és egyszer a kimenő feldolgozásban (ha az **ipforwarding** beállítás meg van adva). Ha egy csomagot el szeretne jutni az A hosztról a B hosztra a G átjárón keresztül, akkor egy engedélyező szabályra van szüksége az alábbiakkal:

- Az A hoszton az **src addr** beállításban adja meg az A értéket, a **dest addr** beállításban pedig a B értéket a kimenő irányban
- Az B hoszton az **src addr** beállításban adja meg az A értéket, a **dest addr** beállításban pedig a B értéket a bejövő irányban

A G átjárón viszont két szabályt is be kell állítani:

1. Az **src addr** beállításban adja meg az A értéket, a **dest addr** beállításban pedig a B értéket a kimenő irányban
2. Az **src addr** beállításban adja meg az A értéket, a **dest addr** beállításban pedig a B értéket a bejövő irányban

A fenti szabályok a következőkkel helyettesíthetők: az **src addr** beállításban adja meg az A értéket, a **dest addr** beállításban a B értéket, iránynak pedig a both értéket. A **both** iránybeállítást tehát olyan átjárókban szokták használni, amelyeknél az **ipforwarding** beállítás no értékre van állítva. A fenti konfiguráció csak az A hosztról a B hosztra a G átjárón keresztül utazó csomagokra vonatkozik. Ha a csomagokat a fordított irányba is továbbítani szeretné (a B hosztról az A hosztra a G átjárón keresztül), akkor ehhez egy másik szabályra van szükség.

Megjegyzés: A **both** irány azt jelzi, hogy a társított szabályt a rendszer a bejövő és kimenő csomagokra is alkalmazza. Ugyanakkor ez nem jelenti azt, hogy a szabályt a rendszer akkor is alkalmazza, ha a forrás- és célcímeket megfordítjuk. Ha például az A szerver szabályában a forráscím *A*, a célcím *B*, az irány pedig **both**, akkor az *A* bejövő csomag *B* forráscímmel és *A* célcímmel nem felel meg ennek a szabálynak. A **both** beállítást általában csomagokat továbbító átjárókon használják.

A szűrőszabályokhoz társíthatók alhálózati maszkok, csoportazonosítók és hoszt-tűzfal-hoszt konfigurációs beállítások. A különféle szűrőszabályokat és szolgáltatásaikat az alábbi szakaszok írják le.

IP szűrők AIX rendszerhez:

Az IPFilter egy szoftvercsomag, amelynek segítségével hálózati cím fordítás (NAT) vagy tűzfal szolgáltatások biztosíthatók.

Az IPFilter 4.1.13 változatú nyílt forrású szoftver át lett írva AIX rendszerre, amely konzisztens az IP szűrő webhelyen látható licenccel (http://coombs.anu.edu.au/~avalon/). Az IPFilter szoftver az AIX bővítőcsomaggal együtt kerül szállításra. Az ipfl installp csomag tartalmazza a man oldalt és a licencet.

AIX operációs rendszeren az IPFilter termék /usr/lib/drivers/ipf kernelkiterjesztésként kerül betöltésre. Az **ipf**, **ipfs**, **ipfstat**, **ipmon** és **ipnat** bináris fájlt szintén tartalmazza a csomag.

A csomag telepítése után futtassa a következő csomagot a kernelkiterjesztés betöltése érdekében:

```
/usr/lib/methods/cfg_ipf -l
```

Futtassa a következő parancsot a kernelkiterjesztés eltávolítása érdekében:

```
/usr/lib/methods/cfg_ipf -u
```

Ne felejtse el engedélyezni az ip továbbítást (ipforwarding hálózati beállítás), ha csomagtovábbítás szükséges. Az IPFilterrel kapcsolatos további információkat, a man oldalakat és GYIK kérdéseket is beleértve az IPFilter webhely tartalmaz (http://coombs.anu.edu.au/~avalon/).

Statikus szűrőszabályok:

A statikus szűrőszabályok szóközzel elválasztott mezőket tartalmaznak.

A statikus szűrőszabály egyes mezőinek nevét az alábbi lista sorolja fel, amelyet az 1. szabály példái követnek zárójelben:

- Rule_number (1)
- Action (permit)
- Source_addr (0.0.0.0)
- Source_mask (0.0.0.0)
- Dest_addr (0.0.0.0)
- Dest_mask (0.0.0.0)
- Source_routing (no)
- Protocol (udp)
- Src_prt_operator (eq)
- Src_prt_value (4001)
- Dst_prt_operator (eq)
- Dst_prt_value (4001)
- Scope (both)
- Direction (both)
- Logging (no)
- Fragment (all packets)
- Tunnel (0)
- Interface (all).

Példa statikus szűrőszabályokra

```
1 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no udp eq 4001 eq 4001 both both no all
   packets 0 all

2 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no ah any 0 any 0 both both no all packets
   0 all

3 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no esp any 0 any 0 both both no all packets
   0 all

4 permit 10.0.0.1 255.255.255.255 10.0.0.2 255.255.255.255 no all any 0 any 0 both
```

```
   outbound no all packets 1 all kimenő forgalom

5 permit 10.0.0.2 255.255.255.255 10.0.0.1 255.255.255.255 no all any 0 any 0 both
   inbound no all packets 1 all


6 permit 10.0.0.1 255.255.255.255 10.0.0.3 255.255.255.255 no tcp lt 1024 eq 514 local
   outbound yes all packets 2 all


7 permit 10.0.0.3 255.255.255.255 10.0.0.1 255.255.255.255 no tcp/ack eq 514 lt 1024
   local inbound yes all packets 2 all


8 permit 10.0.0.1 255.255.255.255 10.0.0.3 255.255.255.255 no tcp/ack lt 1024 lt 1024
   local outbound yes all packets 2 all


9 permit 10.0.0.3 255.255.255.255 10.0.0.1 255.255.255.255 no tcp lt 1024 lt 1024 local
   inbound yes all packets 2 all


10 permit 10.0.0.1 255.255.255.255 10.0.0.4 255.255.255.255 no icmp any 0 any 0 local
   outbound yes all packets 3 all


11 permit 10.0.0.4 255.255.255.255 10.0.0.1 255.255.255.255 no icmp any 0 any 0 local
   inbound yes all packets 3 all


12 permit 10.0.0.1 255.255.255.255 10.0.0.5 255.255.255.255 no tcp gt 1023 eq 21 local
   outbound yes all packets 4 all


13 permit 10.0.0.5 255.255.255.255 10.0.0.1 255.255.255.255 no tcp/ack eq 21 gt 1023 local
   inbound yes all packets 4 all


14 permit 10.0.0.5 255.255.255.255 10.0.0.1 255.255.255.255 no tcp eq 20 gt 1023 local
   inbound yes all packets 4 all


15 permit 10.0.0.1 255.255.255.255 10.0.0.5 255.255.255.255 no tcp/ack gt 1023 eq 20 local
   outbound yes all packets 4 all


16 permit 10.0.0.1 255.255.255.255 10.0.0.5 255.255.255.255 no tcp gt 1023 gt 1023 local
   outbound yes all packets 4 all


17 permit 10.0.0.5 255.255.255.255 10.0.0.1 255.255.255.255 no tcp/ack gt 1023 gt 1023 local
   inbound yes all packets 4 all


18 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 no all any 0 any 0 both both yes all
   packets
```

A fenti példa szabályainak részletes leírása:

1. szabály

A **Szekciókulcs** démonra vonatkozik. A szabály csak az IPv4 szűrőtáblákban látható. A démon a 4001-es portot használja a szekciókulcs frissítését vezérlő csomagokhoz. Az 1. szabály mutatja be a portszámok használatát a szabályokban.

Megjegyzés: Kizárólag naplózási célok érdekében módosítsa a szabályt.

2. és 3. szabály
A Hitelesítési fejléc (AH) és Beágyazott biztonsági kiterjesztés (ESP) protokollok feldolgozásának engedélyezése.

Megjegyzés: Kizárólag naplózási célok érdekében módosítsa a 2. és 3. szabályt.

4. és 5. szabály
Automatikusan előállított szabályok, amelyek a 10.0.0.1 - 10.0.0.2 címek forgalmát az 1. alagúton továbbítják. A 4. szabály a kimenő forgalomra, az 5. a bejövő forgalomra vonatkozik.

Megjegyzés: A 4. szabályban szerepel egy felhasználó által megadott leírás, a *kimenő forgalom*.

6-9. szabályok
Felhasználó által megadott szabályok, amelyek a kimenő rsh, rcp, rdump, rrestore és rdist szolgáltatásokat szűrik a 10.0.0.1 - 10.0.0.3 címek vonatkozásában a 2. alagúton keresztül. A példában a naplózás értéke yes, vagyis az adminisztrátor megfigyelheti ezen szabályok forgalmát.

10. és 11. szabály
Felhasználó által megadott szabálykészlet, amely a bejövő és kimenő icmp szolgáltatásokat szűri a 10.0.0.1 - 10.0.0.4 címek vonatkozásában a 3. alagúton keresztül.

12-17. szabályok
Felhasználó által megadott szűrőszabályok, amelyek a kimenő Fájlátviteli protokoll (FTP) szolgáltatást szűrik a 10.0.0.1 - 10.0.0.5 címeknél a 4. alagúton keresztül.

18. szabály
Ez az automatikusan előállított szabály mindig a tábla végén található. Példánkban minden olyan csomagot engedélyez, amely a többi szűrőszabálynak nem felelt meg. Beállítható oly módon is, hogy a szabályoknak nem megfelelő csomagok tiltottak legyenek.

Az egyes szabályok az **lsfilt** paranccsal önállóan is megtekinthetők az összes mezővel és azok értékeivel. Például:

```
Rule 1:
Rule action          : permit
Source Address       : 0.0.0.0
Source Mask          : 0.0.0.0
Destination Address  : 0.0.0.0
Destination Mask     : 0.0.0.0
Source Routing       : yes
Protocol             : udp
Source Port          : eq  4001
Destination Port     : eq  4001
Scope                : both
Direction            : both
Logging control      : no
Fragment control     : all packets
Tunnel ID number     : 0
Interface            : all
Auto-Generated       : yes
```

A következő lista a szűrőszabályokban megadható paramétereket foglalja össze:

-v IPv4 vagy IPv6

-a Tevékenység:

- **d** Elutasítás
- **p** Engedélyezés

-s Forráscím. IP cím vagy hosztnév lehet.

-m Forrás alhálózati maszk.

-d Célcím. IP cím vagy hosztnév lehet.

-M Cél alhálózati maszk.

-g Forrás útválasztás felügyelete: y vagy n.

-c Protokoll. Az értékek az alábbiak lehetnek: udp, icmp, tcp, tcp/ack, ospf, pip, esp, ah és all.

-o Forrásport vagy ICMP típus művelet.

-p Forrásport vagy ICMP típus érték.

-O Célport vagy ICMP kód művelet.

-P Célport vagy ICMP kód érték.

-r Útválasztás:

- **r** Továbbított csomagok.
- **l** Helyi célú/eredetű csomagok.
- **b** Mindkettő.

-l Naplózás.

- **y** Bekerül a naplóba.
- **n** Nem kerül be a naplóba.

-f Töredezettség.

- **y** Töredékfejlécekre, töredékekre és nem töredékekre érvényes.
- **o** Csak töredékekre és töredékfejlécekre érvényes.
- **n** Nem töredékekre érvényes.
- **h** Csak nem töredékekre és töredékfejlécekre érvényes

-t Alagút azonosító.

-i Csatoló, például tr0 vagy en0.

További információkat a **genfilt** és **chfilt** parancsok leírásánál találhat.

Automatikusan előállított és felhasználó által megadott szűrőszabályok:

Bizonyos szabályokat a rendszer automatikusan előállít az IP biztonság szűrési és alagút kódjának használatához.

Az automatikusan előállított szabályok az alábbi szabálykészleteket tartalmazzák:

- Az IKE IPv4 kulcsok frissítését végző szekciókulcs démon szabályai
- Az AH és ESP csomagok feldolgozására vonatkozó szabályok.

A szűrőszabályok szintén automatikusan előállítottak alagút meghatározásakor. Kézi alagutak esetén az automatikusan előállított szabályok a forrás- és célcímeket, a maszk értékeket és az alagút azonosítóját adják meg. Ezen címek között a teljes forgalom az alagúton keresztül bonyolódik.

IKE alagutak esetén az automatikusan előállított szűrőszabályok a protokollt és portszámokat adják meg az IKE egyeztetés során. Az IKE szűrőszabályok külön táblában találhatók, amelynek keresésére a statikus szűrőszabályok után, de még az automatikusan előállított szabályok előtt kerül sor. Az IKE szűrőszabályok a statikus szűrők táblájában egy alapértelmezett helyen kerülnek beszúrásra, ezeket azonban a felhasználó áthelyezheti.

Az automatikusan előállított szabályok minden forgalmat megengednek az alagútban. A felhasználó által megadott szabályok korlátozásokat támaszthatnak bizonyos forgalommal szemben. A felhasználó által megadott szabályokat az automatikusan előállított szabályok elé kell helyezni, mivel az IP biztonság a csomagra alkalmazható első szabályt alkalmazza. A következő olyan felhasználó által megadott szűrőszabályokra mutat be egy példát, amelyek ICMP művelet alapján szűrik a forgalmat.

```
1 permit 10.0.0.1 255.255.255.255 10.0.0.4 255.255.255.255 no icmp any 8 any 0
   local outbound no all packets 3 all
2 permit 10.0.0.4 255.255.255.255 10.0.0.1 255.255.255.255 no icmp any 0 any 0 local
   inbound no all packets 3 all
3 permit 10.0.0.4 255.255.255.255 10.0.0.1 255.255.255.255 no icmp any 8 any 0 local
   inbound no all packets 3 all
4 permit 10.0.0.1 255.255.255.255 10.0.0.4 255.255.255.255 no icmp any 0 any 0 local
   outbound no all packets 3 all
```

Az egyedülálló alagutak beállításának megkönnyítése érdekében a szűrőszabályok automatikusan létrejönnek az alagutak meghatározásakor. A funkció a **gentun** parancs **-g** paraméterével kapcsolható ki. Az /usr/samples/ipsec/filter.sample fájl egy olyan példafájlt tartalmaz, amely különféle TCP/IP szolgáltatásokra vonatkozóan hoz létre szűrőszabályokat a **genfilt** paranccsal.

Előre meghatározott szűrőszabályok:

Számos előre meghatározott szűrőszabály kerül automatikusan előállításra bizonyos eseményekkel.

előre meghatározott szabály kerül be például a szűrőtáblába egy ipsec_v4 vagy ipsec_v6 eszköz betöltésekor. Alapértelmezésben ez az előre meghatározott szabály minden csomagot engedélyez, ám beállítható úgy is, hogy minden csomagot tiltson.

Megjegyzés: Távoli konfiguráció esetén győződjön meg róla, hogy a tiltási szabály a konfigurálás befejezése előtt nem kerül aktiválásra, ellenkező esetben kizárhatja magát a gépről. Ezt egy alapértelmezett engedélyező szabállyal, vagy egy alagút meghatározásával lehet elkerülni.

Az IPv4 és IPv6 szűrőtáblák egyaránt rendelkeznek előre meghatározott szabállyal. Ezek mindegyike egymástól függetlenül beállítható minden csomag tiltására. Ez megakadályozza, hogy a további szűrőszabályok által kifejezetten engedélyezett csomagokon kívül bármi is áthaladjon. Az előre meghatározott szabályok egyetlen további lehetséges módosítása a csomagok naplózása; ehhez a **chfilt** szabályt kell használni a **-l** kapcsolóval.

Az IKE alagutak támogatásához az IPv4 szűrőtáblába bekerül egy dinamikus szűrőszabály. Ebben a pontban kerülnek beszúrásra a dinamikus szűrőszabályok a szűrőtáblázatba. A pozíciót a felhasználó módosíthatja a szűrőtáblában való felfelé és lefelé mozgatással. Miután az alagútkezelő démon és az **isakmpd** démon elindult az IKE alagutak egyeztetésének biztosítása érdekében, a rendszer automatikusan létrehozza a megfelelő szabályokat a dinamikus szűrőtáblában az IKE üzenetek, illetve AH és ESP csomagok kezeléséhez.

Alhálózati maszkok:

Az alhálózati maszkok használhatók az adott szűrőszabályhoz tartozó azonosítók csoportosítására. A rendszer a maszk értékén és a szűrőszabályban szereplő azonosítón logikai és műveletet hajt végre, majd összehasonlítja az eredményt a csomagban megadott azonosítóval.

A 10.10.10.4 forrás IP címmel és 255.255.255.255 alhálózati maszkkal rendelkező szűrőszabály esetén például a következő IP cím fog pontos egyezést jelenteni:

	Bináris	**Decimális**
Forrás IP cím	1010.1010.1010.0100	10.10.10.4
Alhálózati maszk	11111111.11111111.11111111.11111111	255.255.255.255

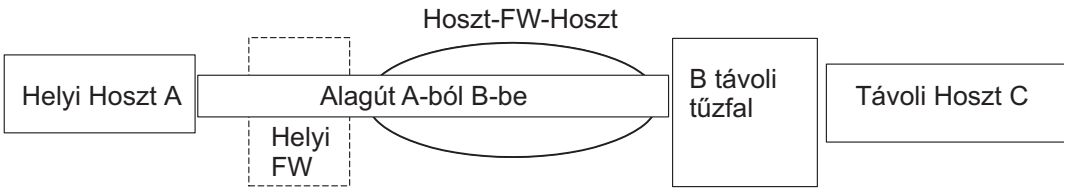
A 10.10.10.x alhálózat 11111111.11111111.11111111.0-ként vagy 255.255.255.0-ként van megadva. Ehhez jön hozzá a bejövő cím, és ez a kombináció kerül összehasonlításra a szűrőszabályban szereplő azonosítóval. Az alhálózati maszk alkalmazása után a 10.10.10.100 például 10.10.10.0 lesz, amely azt jelenti, hogy megfelel a szűrőszabálynak.

A 255.255.255.240 alhálózati maszk esetén a cím utolsó négy bitjén bármi állhat.

Hoszt-tűzfal-hoszt konfiguráció:

Az alagutak hoszt-tűzfal-hoszt konfigurációs beállítása lehetővé teszi alagút létrehozását a helyi hoszt és egy tűzfal között, majd a szükséges szűrőszabályok automatikus előállítását a helyi hoszt és a tűzfal mögötti hoszt közötti megfelelő kommunikációhoz.

Az automatikusan előállított szűrőszabályok a megadott alagúton minden csomagot engedélyeznek a két nem tűzfal hoszt között. A Felhasználói adatcsomag protokoll (UDP), a Hitelesítési fejléc (AH) és Beágyazott biztonsági kiterjesztés (ESP) protokollok alapértelmezett szabályainak már megfelelően kell kezelniük a hoszt és tűzfal közötti kommunikációt. A tűzfalat megfelelően kell beállítani a konfiguráció működéséhez. A tűzfal által igényelt SPI értékek és kulcsok megadásához exportálja a fájlt az alagútból.



14. ábra: Hoszt-tűzfal-hoszt

Ez az ábra mutatja be a hoszt-tűzfal-hoszt konfigurációt. Az A hosztból egy alagút vezet a helyi tűzfalhoz, és ki az Internetre. Ez elmegy a B távoli tűzfalig, majd onnan a C távoli hosztig.

Naplózási szolgáltatások

A hosztok egymással folytatott kommunikációja során az átvitt csomagokat a rendszer naplódémon (syslogd) naplózhatja. Az IP biztonságra vonatkozó más fontos üzenetek is megjelenhetnek itt.

Az adminisztrátorok a naplózási információkat a forgalom elemzése és hibakeresési célból figyelhetik meg. A naplózási szolgáltatások az alábbi lépésekkel állíthatók be.

1. Módosítsa az /etc/syslog.conf fájlt, és adja hozzá a következő bejegyzést:

   ```
   local4.debug var/adm/ipsec.log
   ```

   A local4 szolgáltatás használható a forgalom és az IP biztonsági események feljegyzésére. A naplózásra az operációs rendszer szabványos prioritásai vonatkoznak. A prioritási szintet érdemes mindaddig a hibakeresés szinten tartani, amíg az IP biztonsági alagutakon és szűrőkön áthaladó forgalom nem stabilizálódik.

   Megjegyzés: A szűrőesemények naplózása jelentős terhelést jelenthet az IP biztonsági hoszton, és nagy területet foglalhat el.
2. Mentse el a /etc/syslog.conf file fájlt.
3. Lépjen be a naplófájl számára megadott könyvtárba, és hozzon létre egy üres fájlt a megadott néven. A fenti példánál maradva lépjen be a /var/adm könyvtárba, és adja ki a következő parancsot:

   ```
   touch ipsec.log
   ```
4. Adja ki a **refresh** parancsot a syslogd alrendszernek:

   ```
   refresh -s syslogd
   ```
5. IKE alagutak használata esetén győződjön meg róla, hogy az /etc/isakmpd.conf fájl a kívánt **isakmpd** naplózási szintet adja meg. (Az IKE naplózással kapcsolatos részletes információkat az alábbi rész tartalmaz: “Internet protokoll biztonsági probléma diagnózis” oldalszám: 256.)
6. Ha a hoszt szűrőszabályainak létrehozása során bizonyos szabályoknak megfelelő csomagokat naplózni kíván, akkor a **genfilt** vagy a **chfilt** parancs segítségével állítsa be a szabály *-l* paraméterét **Y**-ra (Yes).
7. Kapcsolja be a csomag naplózást, és indítsa el az **ipsec_logd** démont a következő paranccsal:

   ```
   mkfilt -g start
   ```

   A csomagok naplózását a következő parancs kiadásával állíthatja le:

```
mkfilt -g stop
```

A következő naplófájl minta forgalommal kapcsolatos bejegyzéseket és más IP biztonsági naplóbejegyzéseket tartalmaz:

```
1. Aug 27 08:08:40 host1 : Filter logging daemon ipsec_logd (level 2.20)
   initialized at 08:08:40 on 08/27/97A
2. Aug 27 08:08:46 host1 : mkfilt: Status of packet logging set to Start
   at 08:08:46 on 08/27/97
3. Aug 27 08:08:47 host1 : mktun: Manual tunnel 2 for IPv4, 9.3.97.244, 9.3.97.130
   activated.
4. Aug 27 08:08:47 host1 : mkfilt: #:1 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
   udp eq  4001 eq  4001  both both l=n f=y t=0 e= a=
5. Aug 27 08:08:47 host1 : mkfilt: #:2 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
   ah any 0 any 0  both both l=n f=y t=0 e= a=
6. Aug 27 08:08:47 host1 : mkfilt: #:3 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
   esp any 0 any 0  both both l=n f=y t=0 e= a=
7. Aug 27 08:08:47 host1 : mkfilt: #:4 permit 10.0.0.1 255.255.255.255 10.0.0.2
   255.255.255.255 icmp any 0 any 0  local outbound l=y f=y t=1 e= a=
8. Aug 27 08:08:47 host1 : mkfilt: #:4 permit 10.0.0.2 255.255.255.255 10.0.0.1
   255.255.255.255 icmp any 0 any 0  local inbound l=y f=y t=1 e= a=
9. Aug 27 08:08:47 host1 : mkfilt: #:6 permit 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
    all any 0 any 0  both both l=y f=y t=0 e= a=
10. Aug 27 08:08:47 host1 : mkfilt: Filter support (level 1.00) initialized at
    08:08:47 on 08/27/97
11. Aug 27 08:08:48 host1 : #:6 R:p  o:10.0.0.1 s:10.0.0.1 d:10.0.0.20 p:udp
    sp:3327 dp:53 r:l a:n f:n T:0 e:n l:67
12. Aug 27 08:08:48 host1 : #:6 R:p  i:10.0.0.1 s:10.0.0.20 d:10.0.0.1 p:udp
    sp:53 dp:3327 r:l a:n f:n T:0 e:n l:133
13. Aug 27 08:08:48 host1 : #:6 R:p  i:10.0.0.1 s:10.0.0.15 d:10.0.0.1 p:tcp
    sp:4649 dp:23 r:l a:n f:n T:0 e:n l:43
14. Aug 27 08:08:48 host1 : #:6 R:p  o:10.0.0.1 s:10.0.0.1 d:10.0.0.15 p:tcp
    sp:23 dp:4649 r:l a:n f:n T:0 e:n l:41
15. Aug 27 08:08:48 host1 : #:6 R:p  i:10.0.0.1 s:10.0.0.15 d:10.0.0.1 p:tcp
    sp:4649 dp:23 r:l a:n f:n T:0 e:n l:40
16. Aug 27 08:08:51 host1 : #:4 R:p  o:10.0.0.1 s:10.0.0.1 d:10.0.0.2 p:icmp
    t:8 c:0 r:l a:n f:n T:1 e:n l:84
17. Aug 27 08:08:51 host1 : #:5 R:p  i:10.0.0.1 s:10.0.0.2 d:10.0.0.1 p:icmp
    t:0 c:0 r:l a:n f:n T:1 e:n l:84
18. Aug 27 08:08:52 host1 : #:4 R:p  o:10.0.0.1 s:10.0.0.1 d:10.0.0.2 p:icmp
    t:8 c:0 r:l a:n f:n T:1 e:n l:84
19. Aug 27 08:08:52 host1 : #:5 R:p  i:10.0.0.1 s:10.0.0.2 d:10.0.0.1 p:icmp
    t:0 c:0 r:l a:n f:n T:1 e:n l:84
20. Aug 27 08:32:27 host1 : Filter logging daemon terminating at 08:32:27 on
    08/27/97l
```

A naplóbejegyzéseket a következő szakaszok írják le.

1 A szűrőnaplózási démon aktiválódott.

2 A csomagszűrés naplózása bekapcsolásra került az **mkfilt -g start** paranccsal.

3 Alagút aktiválás; az üzenetben megjelenik az alagút azonosítója, a forráscím, a célcím és az időbélyeg.

4-9 Szűrők aktiválása. A naplózásban megjelenik az összes betöltött szűrőszabály.

10 Szűrők aktiválása.

11-12 Egy hoszt DNS kikeresése.

13-15 Ezek a bejegyzések egy részleges Telnet kapcsolatot mutatnak (a további bejegyzéseket terjedelmi okokból eltávolítottuk).

16-19 Két pingelés.

20 A szűrőnaplózási démon leáll.

A következő példa bemutat egy hosztok közötti 1. fázisú és 2. fázisú egyeztetést a kezdeményező hoszt szemszögéből. (Az **isakmpd** naplózási szint beállítása **isakmp_events**.)

```
1. Dec  6 14:34:42 host1 Tunnel Manager: 0: TM is processing a
   Connection_request_msg
 2. Dec  6 14:34:42 host1 Tunnel Manager: 1: Creating new P1 tunnel object (tid)
 3. Dec  6 14:34:42 host1 isakmpd: 192.168.100.103 >>> 192.168.100.104 ( SA PROPOSAL
   TRANSFORM  )
 4. Dec  6 14:34:42 host1 isakmpd: ::ffff:192.168.100.103 <<< 192.168.100.104 ( SA
   PROPOSAL TRANSFORM  )
 5. Dec  6 14:34:42 host1 isakmpd: Phase I SA Negotiated
 6. Dec  6 14:34:42 host1 isakmpd: 192.168.100.103 >>> 192.168.100.104 ( KE NONCE  )
 7. Dec  6 14:34:42 host1 isakmpd: ::ffff:192.168.100.103 <<< 192.168.100.104 ( KE
   NONCE  )
 8. Dec  6 14:34:42 host1 isakmpd: Encrypting the following msg to send: ( ID HASH
    )
 9. Dec  6 14:34:42 host1 isakmpd: 192.168.100.103 >>> 192.168.100.104 ( Encrypted
   Payloads )
10. Dec  6 14:34:42 host1 isakmpd: ::ffff:192.168.100.103 <<< 192.168.100.104 (
   Encrypted Payloads )
11. Dec  6 14:34:42 host1 Tunnel Manager: 1: TM is processing a P1_sa_created_msg
   (tid)
12. Dec  6 14:34:42 host1 Tunnel Manager: 1:   Received good P1 SA, updating P1
   tunnel (tid)
13. Dec  6 14:34:42 host1 Tunnel Manager: 0: Checking to see if any P2 tunnels need
   to start
14. Dec  6 14:34:42 host1 isakmpd: Decrypted the following received msg: ( ID HASH
    )
15. Dec  6 14:34:42 host1 isakmpd:  Phase I Done !!!
16. Dec  6 14:34:42 host1 isakmpd: Phase I negotiation authenticated
17. Dec  6 14:34:44 host1 Tunnel Manager: 0: TM is processing a
   Connection_request_msg
18. Dec  6 14:34:44 host1 Tunnel Manager: 0: Received a connection object for an
   active P1 tunnel
19. Dec  6 14:34:44 host1 Tunnel Manager: 1: Created blank P2 tunnel (tid)
20. Dec  6 14:34:44 host1 Tunnel Manager: 0: Checking to see if any P2 tunnels need
   to start
21. Dec  6 14:34:44 host1 Tunnel Manager: 1: Starting negotiations for P2 (P2 tid)
22. Dec  6 14:34:45 host1 isakmpd: Encrypting the following msg to send: ( HASH SA
   PROPOSAL TRANSFORM NONCE ID ID  )
23. Dec  6 14:34:45 host1 isakmpd: 192.168.100.103 >>> 192.168.100.104 ( Encrypted
   Payloads )
24. Dec  6 14:34:45 host1 isakmpd: ::ffff:192.168.100.103 <<< 192.168.100.104 (
   Encrypted Payloads )
25. Dec  6 14:34:45 host1 isakmpd: Decrypted the following received msg: ( HASH SA
   PROPOSAL TRANSFORM NONCE ID ID  )
26. Dec  6 14:34:45 host1 isakmpd: Encrypting the following msg to send: ( HASH  )
27. Dec  6 14:34:45 host1 isakmpd: 192.168.100.103 >>> 192.168.100.104 ( Encrypted
   Payloads )
28. Dec  6 14:34:45 host1 isakmpd: Phase II SA Negotiated
29. Dec  6 14:34:45 host1 isakmpd: PhaseII negotiation complete.
30. Dec  6 14:34:45 host1 Tunnel Manager: 0: TM is processing a P2_sa_created_msg
31. Dec  6 14:34:45 host1 Tunnel Manager: 1: received p2_sa_created for an existing
   tunnel as initiator (tid)
32. Dec  6 14:34:45 host1 Tunnel Manager: 1: Filter::AddFilterRules: Created filter
   rules for tunnel
33. Dec  6 14:34:45 host1 Tunnel Manager: 0: TM is processing a List_tunnels_msg
```

A naplóbejegyzéseket a következő szakaszok írják le.

1-2 Az **ike cmd=activate phase=1** parancs kezdeményezi a kapcsolatot.

3-10 Az **isakmpd** démon egyeztet egy 1. fázisú alagutat.

11-12 Az Alagútkezelő kap egy érvényes 1. fázisú biztonsági megegyezést a válaszadótól.

13 Az Alagútkezelő ellenőrzi, hogy az **ike cmd=activate** rendelkezik-e 2. fázisú értékkel. A példában nem.

14-16 Az **isakmpd** démon befejezi az 1. egyeztetési fázist.

17-21 Az **ike cmd=activate phase=2** parancs kezdeményez egy 2. fázisú alagutat.

22-29 Az **isakmpd** démon egyeztet egy 2. fázisú alagutat.

30-31 Az Alagútkezelő kap egy érvényes 2. fázisú biztonsági megegyezést a válaszadótól.

32 Az Alagútkezelő előállítja a dinamikus szűrőszabályokat.

33 Az **ike cmd=list** parancs megjeleníti az IKE alagutakat.

Mezőbejegyzések címkéi:

A naplóbejegyzésekben szereplő mezők lemezterület szempontok miatt rövidített formában kerülnek kiírásra.

Mező	Jelentés
#	A szabály száma, amely alapján a csomag naplózásra került.
R	Szabály típusa **p** Engedélyezés **d** Elutasítás
i/o	A csomag iránya, amikor a szűrő elfogta azt. Megadja a csomaghoz tartozó csatoló IP címét: • Bejövő (i) csomagoknál ez az a csatoló, amelyre a csomag megérkezett. • Kimenő (o) csomagok esetén az IP réteg által meghatározott adapternek kell kezelnie a csomag átvitelét.
s	Megadja a csomag küldőjének IP címét (az IP fejléc alapján).
d	Megadja a csomag címzettjének IP címét (az IP fejléc alapján).
p	Megadja a csomag adatrészében szereplő üzenet létrehozását végző magasszintű protokollt. Szám vagy név lehet, például: udp, icmp, tcp, tcp/ack, ospf, pip, esp, ah vagy all.
sp/t	Megadja a csomag küldőjéhez tartozó a protokoll portszámát (a TCP vagy UDP fejléc alapján). Ha a protokoll ICMP vagy OSPF, akkor a mező helyére **t** kerül, amely az IP típusra utal.
dp/c	Megadja a csomag címzettjéhez tartozó a protokoll portszámát (a TCP vagy UDP fejléc alapján). Ha a protokoll ICMP, akkor a mező helyére **c** kerül, amely az IP kódra utal.
-	Megadja, hogy nincsenek rendelkezésre álló információk.
r	Jelzi, hogy a csomag rendelkezik-e helyi társsal. **f** Továbbított csomagok **l** Helyi csomagok **o** Kimenő **b** Mindkettő
l	Megadja az adott csomag méretét byte-ban.
f	Jelzi, hogy a csomag egy töredék.
T	Megadja az alagút azonosítóját.
i	Megadja, hogy a csomag mely csatolón lépett be.

Internetes kulcscsere naplózása:

Engedélyezheti az Internet kulcscsere események naplózását a SYSLOG szolgáltatásba az **isakmpd** démonnal.

Az **isakmpd** démon esetén a naplózás az **ike cmd=log** parancs segítségével engedélyezhető. Beállíthatja a naplózási szintet az /etc/isakmpd.conf konfigurációs fájlban a **log_level** paraméterrel. A naplózandó információk mennyiségétől függően állítsa a szintet az alábbiak egyikére: *nincs*, *hibák*, *isakmp_események* vagy *információk*.

Ha például a protokoll és megvalósítási információkat kívánja naplózni, akkor adja meg az alábbi paramétert:

```
log_level=INFORMATION
```

Az **isakmpd** démon elindítja a két folyamat egyikét: elküld egy vagy kiértékel egy ajánlást. Az ajánlás elfogadásakor létrejön egy biztonsági megegyezés, és beállításra kerül az alagút. Ha az ajánlás nem kerül elfogadásra, vagy a kapcsolat az egyeztetés befejezése előtt megszűnik, akkor az **isakmpd** démon hibát jelez. A SYSLOG **tmd** bejegyzései

jelzi, hogy az egyeztetés sikerült-e. Egy olyan igazolás hibát okoz, amely érvénytelen és naplózásra került a SYSLOG-ba. A meghiúsult egyeztetés pontos okának meghatározásához tekintse át a /etc/syslog.conf fájlban megadott naplózó fájlt.

A SYSLOG szolgáltatás a napló minden egyes sorához hozzáad egy előtagot, amelyben a dátum, az időpont, a számítógép és a program neve található. A következő példában a számítógép neve googly, a program neve pedig isakmpd:

```
Nov 20 09:53:50 googly isakmpd: ISAKMP_MSG_HEADER
Nov 20 09:53:50 googly isakmpd:  Icookie : 0xef06a77488f25315, Rcookie :0x0000000000000000
Nov 20 09:53:51 googly isakmpd:  Next Payload : 1(SA), Maj Ver : 1, Min Ver : 0
Nov 20 09:53:51 googly isakmpd:  Xchg Type : 2 (ID protected), Flag= 0, Encr : No,COMMIT : No
Nov 20 09:53:51 googly isakmpd:  Msg ID  : 0x00000000
```

Az olvashatóság javítása érdekében a **grep** paranccsal szűrje ki az érdekes naplósorokat (például minden **isakmpd** naplózása) és a **cut** paranccsal távolítsa el az előtagot minden sorról.

Az /etc/isakmpd.conf fájl:

Az **isakmpd** démon beállításait az /etc/isakmpd.conf fájlban adhatja meg.

Az /etc/isakmpd.conf fájlban az alábbi beállítások állnak rendelkezésre.

Naplókonfiguráció
Határozza meg a naplózni kívánt információk mennyiségét. Majd állítsa be a szintet. Az IKE démonok ezen lehetőség segítségével adják meg a naplózás szintjét.

Szintaxis: none | error | isakmp_events | információk

ahol az egyes szintek jelentése az alábbi:

none Nincs naplózás. Ez az alapértelmezett.

error A protokoll és alkalmazás programozási felület (API) hibáinak naplózása.

isakmp_events
IKE protokoll események és hibák naplózása. Ezt a szintet probléma hibakeresése esetén érdemes használni.

information
Protokoll és megvalósítási információk naplózása.

Ismeretlen IP cím egyeztetés
Ezt a lehetőséget YES (igen) vagy NO (nem) értékre állíthatja. Ha igenre állítja, akkor a helyi IKE adatbázisnak tartalmaznia kell egy IP címet minden 1-es fázisú alagútvégponthoz. YES értéket kell megadni a hoszthoz a bejövő fő módú alagút elfogadásához. Az IP cím lehet az elsődleges azonosító vagy egy választható IP cím, amely hozzá van rendelve másik azonosítótípushoz.

Állítsa ezt a lehetőséget NO értékre a bejövő fő módú kapcsolat elfogadásához. Ha a lehetőséget nemre állítja, akkor a hoszt elfogadhatja a kapcsolatot akkor is, ha az IKE adatbázis nem adja meg az IP címet a phase 1 végpontokhoz. Annak érdekében, hogy a hoszt elfogadja a kapcsolatot, igazolás alapú hitelesítést kell használni. Ez lehetővé teszi, hogy a dinamikusan hozzárendelt IP címmel rendelkező hoszt fő módú csatornát kezdeményezzen a géphez.

Ha nem adja meg ezt a paramétert, akkor az alapértelmezett a NO.

Szintaxis: MAIN_MODE_REQUIRES_IP= YES | NO

SOCKS4 szerver konfiguráció
A SOCKS4_PORTNUM lehetőség nem kötelező. Ha nem adja meg, akkor az alapértelmezett 1080-as SOCKS-szerver port kerül felhasználásra. A SOCKS szerver HTTP szerverrel kommunikációhoz használt portérték.

Szintaxis: *mnemonic* = *value*
Ahol a *mneumonic* és *value* a következő lehet:

SOCKS4_SERVER= a szerver nevét adja meg

SOCKS4_PORTNUM= a SOCKS szerver portszámát adja meg

SOCKS4_USERID= felhasználói azonosító

LDAP szerverkonfiguráció

Szintaxis: *mnemonic* = *value*
ahol a *mnemonic* és *value* az alábbi lehet:

LDAP_SERVER= az LDAP szerver nevét adja meg

LDAP_VERSION= az LDAP szerver változata (2 vagy 3 lehet)

LDAP_SERVERPORT= az LDAP szerver portszáma

LDAP_SEARCHTIME= klienskeresés időtúllépési értéke

CRL lehívási sorrend

A lehetőség megadja, hogy a HTTP vagy LDAP szerver kerül először lekérdezésre, ha mindkét szerver be van állítva. A CRL_FETCH_ORDER lehetőség nem kötelező. Az alapértelmezett lehívási sorrendben a HTTP az első, a következő az LDAP, attól függően, hogy a HTTP és LDAP szerver is be van-e állítva.

Szintaxis: CRL_FETCH_ORDER= *protocol#*, *protocol#*
ahol a *protocol#* HTTP vagy LDAP lehet.

IKEv1 és IKEv2 portspecifikáció

Ez a karaktersorozat megadja az **isakmpd** (IKEv1) és az **ikev2d** (IKEv2) démon által használt portokat. Az **iked** démon (az IKE üzenetközvetítő démon) kikeresi ezt a bejegyzést, majd elindítja az **isakmpd** és **ikev2d** démont a megfelelő portokon.

Szintaxis: v1=port-natport,v2=port-natport

Internet protokoll biztonsági probléma diagnózis

Ez a szakasz néhány tanácsot és tippet ad, amelyek hasznosak lehetnek, ha hibába ütközött.

Naplózás beállítása az IPSec első beállításakor. A naplók nagyon hasznosak a szűrőknél és alagutaknál történtek meghatározásában. (A naplózásról részletesen a “Naplózási szolgáltatások” oldalszám: 251 szakasz ír.)

A futó IP biztonsági démonok meghatározásához adja ki a következő parancsot:

```
ps -ef
```

A következő démonok IP biztonsághoz vannak rendelve: **tmd**, **iked**, **isakmpd**, **ikev2d**, **cpsd**.

Megjegyzés: Ha az IKEv1 és IKEv2 is be van állítva, akkor az **iked** démon fut. Ellenkező esetben vagy az **iskmpd** vagy az **ikev2d** démon fut. Ezt a konfigurációt az **/etc/isakmpd.conf** fájl tartalmazza.

Kézi alagút hibáinak elhárítása:

Az alábbiakban számos lehetséges alagúthibával kapcsolatos leírás és azok megoldását találja.

Hiba	Lehetséges probléma és megoldás
Az **mktun** parancs kiadása a következő hibát eredményezi: insert_tun_man4(): write failed : The requested resource is busy.	Probléma: az aktiválni próbált alagút már aktív, vagy ütköző SPI értékeket ad meg. Helyreállítás: Adja ki az **rmtun** parancsot a leállításhoz, majd az **mktun** parancsot az aktiváláshoz. Nézze meg, hogy a hibát okozó alagút SPI értéke megegyezik-e egy másik aktív alagút értékével. Minden alagútnak egyedi SPI értékekkel kell rendelkeznie.

Az **mktun** parancs kiadása a következő hibát eredményezi:
 	
Device ipsec_v4 is in Defined status.
 	
Tunnel activation for IP Version 4 not performed.
	Probléma: Nem tette elérhetővé az IP biztonsági eszközt.
	
	Helyreállítás: Adja ki a következő parancsot:
	`mkdev -l ipsec -t 4`
	
	Elképzelhető, hogy a **-t** paraméter értékét 6-ra kell módosítani, ha ugyanezt a hibát IPv6 alagút aktiválása esetén kapja. Az eszközöknek elérhető állapotban kell lenniük. Az IP biztonsági eszköz állapotának ellenőrzéséhez adja ki a következő parancsot:
	`lsdev -Cc ipsec`

A **gentun** parancs kiadása a következő hibát eredményezi: Invalid Source IP address	Probléma: Nem adott meg érvényes IP címet forráscímként. Helyreállítás: IPv4 alagutak esetén ellenőrizze, hogy megadta-e a helyi számítógép valamelyik rendelkezésre álló IPv4 címét. Alagutak előállításakor hosztneveket nem adhat meg forráscímként, csak célcímként. IPv6 alagutak esetén ellenőrizze, hogy rendelkezésre álló IPv6 címet adott-e meg. Ha kiadta a netstat -in parancsot és nem láthatók IPv6 címek, akkor futtassa az /usr/sbin/autoconf6 (csatoló) parancsot az IPv6 automatikus (MAC cím alapján végzett) beállításához, vagy használja az **ifconfig** parancsot egy cím kézi hozzárendeléséhez.
A **gentun** parancs kiadása a következő hibát eredményezi: Invalid Source IP address	Probléma: Nem adott meg érvényes IP címet forráscímként. Helyreállítás: IPv4 alagutak esetén ellenőrizze, hogy megadta-e a helyi számítógép valamelyik rendelkezésre álló IPv4 címét. Alagutak előállításakor hosztneveket nem adhat meg forráscímként, csak célcímként. IPv6 alagutak esetén ellenőrizze, hogy rendelkezésre álló IPv6 címet adott-e meg. Ha kiadta a netstat -in parancsot és nem láthatók IPv6 címek, akkor futtassa az /usr/sbin/autoconf6 (csatoló) parancsot az IPv6 automatikus (MAC cím alapján végzett) beállításához, vagy használja az **ifconfig** parancsot egy cím kézi hozzárendeléséhez.
Az **mktun** parancs kiadása a következő hibát eredményezi: insert_tun_man4(): write failed : A system call received a parameter that is not valid.	Probléma: Az alagút előállítása érvénytelen ESP és AH kombinációval, vagy az új fejlécformátum nélkül történt. Helyreállítás: Tekintse meg, hogy a kérdéses alagút milyen hitelesítési algoritmusokat használ. Ne feledje, hogy a HMAC_MD5 és HMAC_SHA algoritmusok az új fejlécformátum használatát igénylik. Az új fejlécformátum az **ips4_basic** SMIT gyorseléréssel vagy a **chtun** parancs *-z* paraméterével módosítható. Ne feledkezzen meg arról sem, hogy a DES_CBC_4 nem használható az új fejlécformátummal.
Az IP biztonság használatára tett kísérlet a következő hibát eredményezi: The installed bos.crypto is back level and must be updated.	Probléma: A bos.net.ipsec.* fájlok frissítésre kerültek, de a megfelelő bos.crypto.* fájlok nem. Helyreállítás: Frissítse a bos.crypto.* fájlokat a bos.net.ipsec.* fájloknak megfelelő szintre.

Internet kulcscsere alagúthibák hibáinak elhárítása:

A következő szakasz az Internet kulcscsere (IKE) alagutak használatával kapcsolatos lehetséges hibákat sorolja fel.

Internet kulcscsere alagút folyamatfolyam:

A fejezet az Internet kulcscsere alagút folyamatfolyamát írja le.

Az IKE alagutak az **ike** parancs és az alábbi démonok kommunikációja által kerülnek beállításra:

tmd Alagútkezelő démon

iked Az IKE közvetítő démon (csak akkor aktív, ha az IKEv1 és az IKEv2 démon egyaránt be van állítva a rendszeren)

isakmpd
IKEv1 démon

ikev2d IKEv2 démon

cpsd Igazolás proxy démon

Az IKE alagutak megfelelő beállításához a **tmd** és **isakmpd** démonnak futnia kell. Ha az IP biztonság beállítása előírja a rendszertöltés utáni indulást, akkor a démonok automatikusan elindulnak. Ellenkező esetben a következő paranccsal kell elindítani azokat:

```
startsrc -g ike
```

Az Alagútkezelő adja át a kéréseket az **isakmpd** parancsnak az alagút elindításához. Ha az alagút már létezik vagy érvénytelen (például egy érvénytelen távoli cím miatt), akkor hibát jelent. Az egyeztetés megkezdése után a befejezés a hálózati késleltetés miatt eltarthat egy darabig. Az **ike cmd=list** parancs megjeleníti az alagút állapotát, amelyből megállapítható, hogy az egyeztetés sikerült-e. Emellett az alagútkezelő a syslog naplóba naplózza a hibakeresés, esemény és információ szintű eseményeket, amelyek segítségével nyomon követhető az egyeztetés folyamata.

A sorrend a következő:

1. Az **ike** paranccsal alagút kezdeményezése.
2. A **tmd** démon átadja az **isakmpd** démonnak a kulcskezelésre (1. fázis) vonatkozó kapcsolati kérést.
3. Az **isakmpd** démon az SA létrejött vagy egy hibaüzenettel válaszol.
4. A **tmd** démon átadja az **isakmpd** démonnak az adatkezelésre (2. fázis) vonatkozó kapcsolati kérést.
5. Az **isakmpd** démon az SA létrejött vagy egy hibaüzenettel válaszol.
6. Az alagút paraméterei bekerülnek a kernel alagút gyorsítótárába.
7. A szűrőszabályok bekerülnek a kernel dinamikus szűrőtáblájába.

Ha a számítógép válaszadóként tevékenykedik, akkor az **isakmpd** démon értesíti a **tmd** alagútkezelő démont, hogy egy alagút sikeresen egyeztetésre került, és az új alagút bekerül a kernelbe. Bizonyos helyzetekben a folyamat a 3. lépéstől tart a 7. lépésig oly módon, hogy a **tmd** démon nem ad ki kapcsolati kéréseket.

Hasznos tartalom naplózás elemzése funkció:

A két végpont közötti biztonsági megegyezés (SA) kialakítása IKE üzenetek cseréjével történik. A hasznos tartalom elemzési funkció teszi emberi értelmezésre alkalmassá az üzeneteket.

A hasznos tartalom naplózás elemzése az /etc/isakmpd.conf fájl módosításával engedélyezhető. Az /etc/isakmpd.conf fájl naplózási bejegyzése a következőhöz hasonlít:

```
information
```

A csomagtartalom értelmező által naplózott IKE csomagtartalmak az IKE üzenetektől függnek. Ilyen például az SA tartalom, a kulcscsere tartalom, az igazolási kérés tartalom, az igazolás tartalom és az aláírás tartalom. A következő példa egy csomagtartalom elemző napló, amelyben egy ISAKMP_MSG_HEADER fejlécet 5 hasznos tartalom követ:

```
ISAKMP_MSG_HEADER
        Icookie : 0x9e539a6fd4540990, Rcookie : 0x0000000000000000
        Next Payload : 1(SA), Maj Ver : 1, Min Ver : 0
        Xchg Type : 4 (Aggressive), Flag= 0, Encr : No,COMMIT : No
        Msg ID  : 0x00000000
        len     : 0x10e(270)
SA Payload:
        Next Payload : 4(Key Exchange), Payload len : 0x34(52)
        DOI          : 0x1(INTERNET)
        bitmask      : 1(SIT_IDENTITY_ONLY
   Proposal Payload:
        Next Payload : 0(NONE), Payload len : 0x28(40)
        Proposal # : 0x1(1), Protocol-ID : 1(ISAKMP)
        SPI size : 0x0(0), # of Trans : 0x1(1)
```

```
   Transform Payload:
        Next Payload : 0(NONE), Payload len : 0x20(32)
        Trans # : 0x1(1), Trans.ID : 1(KEY_IKE)
        Attr : 1(Encr.Alg     ), len=0x2(2)
        Value=0x1(1),(DES-cbc)
        Attr : 2(Hash Alg     ), len=0x2(2)
        Value=0x1(1),(MD5)
        Attr : 3(Auth Method  ), len=0x2(2)
        Value=0x3(3),(RSA Signature)
        Attr : 4(Group Desc   ), len=0x2(2)
        Value=0x1(1),(default 768-bit MODP group)
        Attr : 11(Life Type    ), len=0x2(2)
        Value=0x1(1),(seconds)
        Attr : 12(Life Duration), len=0x2(2)
        Value=0x7080(28800)
Key Payload:
        Next Payload : 10(Nonce), Payload len : 0x64(100)

        Key Data :
        33 17 68 10 91 1f ea da 38 a0 22 2d 84 a3 5d 5d
        a0 e1 1f 42 c2 10 aa 8d 9d 14 0f 58 3e c4 ec a3
        9f 13 62 aa 27 d8 e5 52 8d 5c c3 cf d5 45 1a 79
        8a 59 97 1f 3b 1c 08 3e 2a 55 9b 3c 50 cc 82 2c
        d9 8b 39 d1 cb 39 c2 a4 05 8d 2d a1 98 74 7d 95
        ab d3 5a 39 7d 67 5b a6 2e 37 d3 07 e6 98 1a 6b

Nonce Payload:
        Next Payload : 5(ID), Payload len : 0xc(12)

        Nonce Data:
        6d 21 73 1d dc 60 49 93
ID Payload:
        Next Payload : 7(Cert.Req), Payload len : 0x49(73)
        ID type      : 9(DER_DN), Protocol : 0, Port = 0x0(0)
Certificate Request Payload:
        Next Payload : 0(NONE), Payload len : 0x5(5)
        Certificate Encoding Type: 4(X.509 Certificate - Signature)
```

Minden egyes tartalomban található egy **Következő hasznos tartalom** mező, amely az aktuális tartalom után következő tartalomra mutat. Ha az aktuális tartalom utolsó az IKE üzenetben, akkor a **Következő tartalom** mező értéke nulla.

A példában megadott valamennyi tartalom információkat hordoz a folyamatban lévő egyeztetésekről. Az SA tartalomban található például az ajánlás és az átalakítás, amely meghatározza a kezdeményező által a válaszadónak felajánlott titkosítási algoritmust, hitelesítési módot, kivonatkészítési algoritmust és SA időtartamot.

Emellett az SA tartalomban található legalább egy ajánlás és átalakítás is. Az ajánlás tartalom **Következő tartalom** mezője vagy 0, ha ez az egyetlen ajánlás tartalom, vagy 2, ha még egy ajánlás követi. Hasonlóan, az átalakítás **Következő tartalom** mezője 0, ha ez az egyetlen átalakítás, vagy 3, ha még egy átalakítás követi; ez látható az alábbi példán is:

```
ISAKMP_MSG_HEADER
        Icookie : 0xa764fab442b463c6, Rcookie : 0x0000000000000000
        Next Payload : 1(SA), Maj Ver : 1, Min Ver : 0
        Xchg Type : 2 (ID protected), Flag= 0, Encr : No,COMMIT : No
        Msg ID  : 0x00000000
        len     : 0x70(112)
SA Payload:
        Next Payload : 0(NONE), Payload len : 0x54(84)
        DOI          : 0x1(INTERNET)
        bitmask      : 1(SIT_IDENTITY_ONLY
   Proposal Payload:
        Next Payload : 0(NONE), Payload len : 0x48(72)
        Proposal # : 0x1(1), Protocol-ID : 1(ISAKMP)
        SPI size : 0x0(0), # of Trans : 0x2(2)
```

```
   Transform Payload:
        Next Payload : 3(Transform), Payload len : 0x20(32)
        Trans # : 0x1(1), Trans.ID : 1(KEY_IKE)
        Attr : 1(Encr.Alg     ), len=0x2(2)
        Value=0x5(5),(3DES-cbc)
        Attr : 2(Hash Alg     ), len=0x2(2)
        Value=0x1(1),(MD5)
        Attr : 3(Auth Method  ), len=0x2(2)
        Value=0x1(1),(Pre-shared Key)
        Attr : 4(Group Desc   ), len=0x2(2)
        Value=0x1(1),(default 768-bit MODP group)
        Attr : 11(Life Type    ), len=0x2(2)
        Value=0x1(1),(seconds)
        Attr : 12(Life Duration), len=0x2(2)
        Value=0x7080(28800)
   Transform Payload:
        Next Payload : 0(NONE), Payload len : 0x20(32)
        Trans # : 0x2(2), Trans.ID : 1(KEY_IKE)
        Attr : 1(Encr.Alg     ), len=0x2(2)
        Value=0x1(1),(DES-cbc)
        Attr : 2(Hash Alg     ), len=0x2(2)
        Value=0x1(1),(MD5)
        Attr : 3(Auth Method  ), len=0x2(2)
        Value=0x1(1),(Pre-shared Key)
        Attr : 4(Group Desc   ), len=0x2(2)
        Value=0x1(1),(default 768-bit MODP group)
        Attr : 11(Life Type    ), len=0x2(2)
        Value=0x1(1),(seconds)
        Attr : 12(Life Duration), len=0x2(2)
        Value=0x7080(28800)
```

A hasznos tartalom elemző napló IKE üzenet fejléce megjeleníti az adatcsere típusát (elsődleges vagy agresszív mód), a teljes üzenet hosszát, az üzenet azonosítóját, stb.

Az igazolási kérés tartalom kér igazolást a válaszadótól. A válaszadó az igazolást külön üzenetben küldi. Az alábbi példa mutatja be az SA egyeztetés részeként küldött igazolás tartalmat és aláírás tartalmat. Az igazolás és aláírás adatok hexadecimális formában kerülnek kiírásra.

```
ISAKMP_MSG_HEADER
        Icookie : 0x9e539a6fd4540990, Rcookie : 0xc7e0a8d937a8f13e
        Next Payload : 6(Certificate), Maj Ver : 1, Min Ver : 0
        Xchg Type : 4 (Aggressive), Flag= 0, Encr : No,COMMIT : No
        Msg ID  : 0x00000000
        len     : 0x2cd(717)
Certificate Payload:

        Next Payload : 9(Signature), Payload len : 0x22d(557)
        Certificate Encoding Type: 4(X.509 Certificate - Signature)
        Certificate: (len 0x227(551) in bytes
        82 02 24 30 82 01 8d a0 03 02 01 02 02 05 05 8e
        fb 3e ce 30 0d 06 09 2a 86 48 86 f7 0d 01 01 04
        05 00 30 5c 31 0b 30 09 06 03 55 04 06 13 02 46
        49 31 24 30 22 06 03 55 04 0a 13 1b 53 53 48 20
        43 6f 6d 6d 75 6e 69 63 61 74 69 6f 6e 73 20 53
        65 63 75 72 69 74 79 31 11 30 0f 06 03 55 04 0b
        13 08 57 65 62 20 74 65 73 74 31 14 30 12 06 03
        55 04 03 13 0b 54 65 73 74 20 52 53 41 20 43 41
        30 1e 17 0d 39 39 30 39 32 31 30 30 30 30 30 30
        5a 17 0d 39 39 31 30 32 31 32 33 35 39 35 39 5a
        30 3f 31 0b 30 09 06 03 55 04 06 13 02 55 53 31
        10 30 0e 06 03 55 04 0a 13 07 49 42 4d 2f 41 49
        58 31 1e 30 1c 06 03 55 04 03 13 15 62 61 72 6e
        65 79 2e 61 75 73 74 69 6e 2e 69 62 6d 2e 63 6f
        6d 30 81 9f 30 0d 06 09 2a 86 48 86 f7 0d 01 01
        01 05 00 03 81 8d 00 30 81 89 02 81 81 00 b2 ef
        48 16 86 04 7e ed ba 4c 14 d7 83 cb 18 40 0a 3f
```

```
        55 e9 ad 8f 0f be c5 b6 6d 19 ec de 9b f5 01 a6
        b9 dd 64 52 34 ad 3d cd 0d 8e 82 6a 85 a3 a8 1c
        37 e4 00 59 ce aa 62 24 b5 a2 ea 8d 82 a3 0c 6f
        b4 07 ad 8a 02 3b 19 92 51 88 fb 2c 44 29 da 72
        41 ef 35 72 79 d3 e9 67 02 b2 71 fa 1b 78 13 be
        f3 05 6d 10 4a c7 d5 fc fe f4 c0 b8 b8 fb 23 70
        a6 4e 16 5f d4 b1 9e 21 18 82 64 6d 17 3b 02 03
        01 00 01 a3 0f 30 0d 30 0b 06 03 55 1d 0f 04 04
        03 02 07 80 30 0d 06 09 2a 86 48 86 f7 0d 01 01
        04 05 00 03 81 81 00 75 a4 ee 9c 3a 18 f2 de 5d
        67 d4 1c e4 04 b4 e5 b8 5e 9f 56 e4 ea f0 76 4a
        d0 e4 ee 20 42 3f 20 19 d4 25 57 25 70 0a ea 41
        81 3b 0b 50 79 b5 fd 1e b6 0f bc 2f 3f 73 7d dd
        90 d4 08 17 85 d6 da e7 c5 a4 d6 9a 2e 8a e8 51
        7e 59 68 21 55 4c 96 4d 5a 70 7a 50 c1 68 b0 cf
        5f 1f 85 d0 12 a4 c2 d3 97 bf a5 42 59 37 be fe
        9e 75 23 84 19 14 28 ae c4 c0 63 22 89 47 b1 b6
        f4 c7 5d 79 9d ca d0
Signature Payload:
        Next Payload : 0(NONE), Payload len : 0x84(132)

        Signature: len 0x80(128) in bytes
        9d 1b 0d 90 be aa dc 43 95 ba 65 09 b9 00 6d 67
        b4 ca a2 85 0f 15 9e 3e 8d 5f e1 f0 43 98 69 d8
        5c b6 9c e2 a5 64 f4 ef 0b 31 c3 cb 48 7c d8 30
        e3 a2 87 f4 7c 9d 20 49 b2 39 00 fa 8e bf d9 b0
        7d b4 8c 4e 19 3a b8 70 90 88 2c cf 89 69 5d 07
        f0 5a 81 58 2e 15 40 37 b7 c8 d6 8c 5c e2 50 c3
        4d 19 7e e0 e7 c7 c2 93 42 89 46 6b 5f f8 8b 7d
        5b cb 07 ea 36 e5 82 9d 70 79 9a fe bd 6c 86 36
```

Digitális igazolás és aláírás móddal kapcsolatos problémák:

Az alábbi rész megjeleníti a lehetséges digitális igazolás és aláírás mód problémákat és biztosítja a megfelelő megoldásokat.

Hiba	**Lehetséges probléma és megoldás**
Hiba: A **cpsd** (Igazolás proxy szerver démon) nem indul el. A naplófájlban a következőhöz hasonló bejegyzés található: Sep 21 6:02:00 ripple CPS[19950]: Init():Lo adCaCerts() failed, rc =-12	Probléma: Az igazolás adatbázis nincs megnyitva vagy nem található. Helyreállítás: Győződjön meg róla, hogy a Kulcskezelés igazolás adatbázisai megtalálhatók a /etc/security könyvtárban. Az adatbázist az ikekey.crl, ikekey.kdb, ikekey.rdb és ikekey.sth alkotja. Ha csak az ikekey.sth fájl hiányzik, akkor a Jelszó tárolása beállítás a Kulcskezelés adatbázis létrehozásakor nem lett kiválasztva. Ahhoz, hogy az IP biztonság működjön digitális igazolásokkal, a jelszót tárolni kell. (További információkat a Kulcsadatbázis létrehozása szakaszban talál.)

Hiba	Lehetséges probléma és megoldás

Hiba: A Kulcskezelés a következő hibát adja egy igazolás fogadásakor:
 	
Invalid Base64-encoded data was found
	Probléma: Az igazolás fájlban felesleges adatok találhatók, illetve az adatok elvesztek vagy megsérültek.
	
	Helyreállítás: A 'DER' kódolású igazolásnak a következő sorok között kell lennie (lásd lejjebb). A BEGIN CERTIFICATE és END CERTIFICATE karaktersorozatokat nem előzhetik meg vagy követhetik más karakterek.
	`-----BEGIN CERTIFICATE-----`
	`MIICMTCCAZqgAwIBAgIFFKZtANowDQYJKoZIhvcNAQEFBQAwXDELMAkGA1UEBhMC`
	`RkkxJDAiBgNVBAoTG1NTSCBDb21tdW5pY2F0aW9ucyBTZWN1cml0eTERMA8GA1UE`
	`CxMIV2ViIHRlc3QxFDASBgNVBAMTC1Rlc3QgUlNBIENBMB4XDTk5MDkyMTAwMDAw`
	`MFoXDTk5MTAyMTIzNTk1OVowOzELMAkGA1UEBhMCVVMxDDAKBgNVBAoTA0lCTTEe`
	`MBwGA1UEAxMVcmlwcGxlLmF1c3Rpbi5pYm0uY29tMIGfMA0GCSqGSIb3DQEBAQUA`
	`A4GNADCBiQKBgQC5EZqo6n7tZrpAL6X4L7mf4yXQSm+m/NsJLhp6afbFpPvXgYWC`
	`wq4pvOtvxgum+FHrE0gysNjbKkE4Y6ixC9PGGAKHnhM3vrmvFjnl1G6KtyEz58Lz`
	`BWW39QS6NJ1LqqP1nT+y3+Xzvfv8Eonqzno8mglCWMX09SguLmWoU1PcZQIDAQAB`
	`oyAwHjALBgNVHQ8EBAMCBaAwDwYDVR0RBAgwBocECQNhhzANBgkqhkiG9w0BAQUF`
	`AOBgQA6bgp4Zay34/fyAlyCkNNAYJRrN3Vc4NHN7IGjUziN6jK5UyB5zL37FERW`
	`hT9ArPLzK7yEZs+MDNvB0bosyGWEDYPZr7EZHhYcoBP4/cd0V5rBFmA8Y2gUthPi`
	`Ioxpi4+KZGHYyLqTrm+8Is/DVJaQmCGRPynHK35xjT6WuQtiYg==`
	`-----END CERTIFICATE-----`
	
	A probléma felismeréséhez és megoldásához a következő lehetőségek nyújthatnak segítséget.
	• Ha az adatok elvesztek vagy megsérültek, akkor hozza létre ismét az igazolást.
	• Egy ASM.1 elemző (az Interneten elérhető) segítségével ellenőrizze az igazolás érvényességét.

Hiba: A Kulcskezelés a következő hibát adja egy személyes igazolás fogadásakor: No request key was found for the certificate	Probléma: A fogadott személyes igazoláshoz nem létezik személyes igazolási kérés. Helyreállítás: Hozza létre ismét a személyes igazolási kérést, és kérjen egy új igazolást.
Hiba: Egy IKE egyeztetés meghiúsul, a naplófájlba pedig a következőhöz hasonló bejegyzés kerül: inet_cert_service:: channelOpen(): clientInitIPC():error,rc =2 (No such file or directory)	Probléma: A **cpsd** nem fut vagy leállt. Helyreállítás: Indítsa el az IP biztonságot, ami elindítja a megfelelő démonokat.
Hiba: Egy IKE egyeztetés meghiúsul, a naplófájlba pedig a következőhöz hasonló bejegyzés kerül: CertRepo::GetCertObj: DN Does Not Match: ("/C=US/O=IBM/CN=ripple.austin.ibm.com")	Probléma: Az IKE alagút meghatározásakor megadott X.500 megkülönböztetett név nem egyezik a személyes igazolásban szereplő X.500 megkülönböztetett névvel. Helyreállítás: Módosítsa az IKE alagút meghatározását az igazolásban szereplő megkülönböztetett névnek megfelelően.

Nyomkövetési szolgáltatások:

A nyomkövetés egy hibakeresési szolgáltatás a kernelesemények nyomkövetéséhez. A nyomkövetések segítségével pontosabb információkat lehet kapni a kernel szűrési és alagútkezelési kódjában bekövetkező eseményekről vagy hibákról.

A SMIT IP biztonság nyomkövetési szolgáltatása az IP biztonság további beállításai menüből érhető el. A nyomkövetés által elfogott információk között egyebek között hibákra, szűrőkre, szűrő információkra, alagutakra, alagút információkra, beágyazásra/kibontásra, beágyazási információkra, titkosításra és titkosítási információkra lelhetünk. Tervezéséből adódóan a hiba nyomkövetés biztosítja a legfontosabb információkat. Az információs nyomkövetés szintén biztosít kritikus információkat, viszont hatással lehet a rendszer teljesítményére. Ez a nyomkövetés információkat biztosít a problémáról és akkor is szükséges, amikor elmagyarázza a problémát a szerviz munkatársának.

A nyomkövetés engedélyezéséhez állítsa be az IPSec eszközöket és állítsa minden egyes IPSec részösszetevő nyomkövetési szintjét 7-re, hogy hasznos kernel nyomkövetési adatokat állítson elő. Ha nincsenek beállítva IPSec

eszközök, akkor az összetevő-nyomkövetést vezérlő parancs nem listázza az IP biztonsági protokollhoz kapcsolódó bejegyzéseket. Az IPSec nyomkövetés indításához használja a **smit ips4_start** (IPv4 esetén), illetve a **smit ips6_start** (IPv6 esetén) SMIT gyorselérést.

Megjegyzés: Ha az IPSec összetevő nyomkövetés helytelenül van beállítva, akkor a lementett nyomkövetések üresek lesznek.

Kernel nyomkövetési adatok lementéséhez tegye a következőket:

1. Kérdezze le az összes összetevőt a jelenlegi nyomkövetési szint beállítások megjelenítéséhez:
   ```
   # ctctrl -q
   ```
2. Ellenőrizze az IPSec összetevőt és részösszetevőket. Az összetevők kezdetben az alábbiak szerint jelennek meg az alapértelmezett 3-as nyomkövetési szinttel. Az összetevők kezdeti alapértelmezett nyomkövetési szintjének megjelenítéséhez írja be a következőt:
   ```
   # ctctrl -q -c ipsec -r
   ```

Összetevő neve	Van álneve	Memória nyomkövetés/szint	Rendszer nyomkövetés/szint	Pufferméret/foglalás
ipsec	NO	ON/3	ON/3	40960/YES
.capsulate	NO	ON/3	ON/3	10240/YES
.filter	NO	ON/3	ON/3	10240/YES
.tunnel	NO	ON/3	ON/3	10240/YES

3. Növelje az IPSec és a hozzá tartozó részösszetevők nyomkövetési szintjét 7-re, hogy támogatott legyen a kernel nyomkövetés. Ehhez írja be a következőt:
   ```
   # ctctrl systracelevel=7 -c ipsec -r
   ```
4. Lekérdezéssel ellenőrizze, hogy megváltozott az IPSec és a hozzá tartozó részösszetevők nyomkövetési szintje. Ehhez írja be a következőt:
   ```
   # ctctrl -q -c ipsec -r
   ```

Összetevő neve	Van álneve	Memória nyomkövetés/szint	Rendszer nyomkövetés/szint	Pufferméret/foglalás
ipsec	NO	ON/3	ON/7	40960/YES
.capsulate	NO	ON/3	ON/7	10240/YES
.filter	NO	ON/3	ON/7	10240/YES
.tunnel	NO	ON/3	ON/7	10240/YES

A nyomkövetési szolgáltatás eléréséhez használja a **smit ips4_tracing** (IPv4) vagy **smit ips6_tracing** (IPv6) SMIT gyorselérést. A **smit ips4_tracing**, **smit ips6_tracing** vagy a parancssori nyomkövetési eszköz útján beállított kernel nyomkövetés érvényes IPSec nyomkövetési adatokat ad.

ipsecstat parancs:

Az **ipsecstat** parancs segítségével megjelenítheti az IP biztonsági eszközök állapotát, az IP biztonság kriptográfiai algoritmusokat és az IP biztonság csomagok statisztikáját.

Az **ipsecstat** parancs kiadása előállítja a következő példajelentést, amely a következőket jeleníti meg: az IP biztonsági eszközök elérhető állapotban vannak, három hitelesítési és titkosítási algoritmus van telepítve, emellett megjelennek a csomag tevékenységre és hibákra vonatkozó statisztikai adatok. Ezek az információk hasznosak lehetnek az IP biztonság hatálya alá tartozó forgalom hibaelhárításakor, illetve a problémák helyének meghatározásakor.

```
IP Security Devices:
ipsec_v4 Available
ipsec_v6 Available

Authentication Algorithm:
HMAC_MD5 -- Hashed MAC MD5 Authentication Module
```

```
HMAC_SHA -- Hashed MAC SHA Hash Authentication Module
KEYED_MD5 -- Keyed MD5 Hash Authentication Module

Encryption Algorithm:
CDMF -- CDMF Encryption Module
DES_CBC_4 -- DES CBC 4 Encryption Module
DES_CBC_8 -- DES CBC 8 Encryption Module
3DES_CBC -- Triple DES CBC Encryption Module

IP Security Statistics -
Total incoming packets:  1106
Incoming AH packets:326
Incoming ESP packets:  326
Srcrte packets allowed:  0
Total outgoing packets:844
Outgoing AH packets:527
Outgoing ESP packets:  527
Total incoming packets dropped:  12
  Filter denies on input:  12
  AH did not compute: 0
  ESP did not compute:0
  AH replay violation:0
  ESP replay violation:  0
Total outgoing packets dropped:0
  Filter denies on input:0
Tunnel cache entries added: 7
Tunnel cache entries expired:  0
Tunnel cache entries deleted:  6
```

Megjegyzés: Nincs szükség CDMF használatára, mert a DES már globálisan elérhető. A CDMF algoritmust használó alagutakat át kell állítani a DES vagy tripla DES használatára.

IP biztonsági szolgáltatás hivatkozás

Az IP biztonsághoz vannak parancsok és metódusok. Az IKE alagutakat, a szűrőket és az előzetesen megosztott kulcsokat is átveheti.

Parancsok listája:

Az alábbi táblázat a parancsok listáját tartalmazza.

Parancs	Rendeltetés
ike cmd=activate	Internet kulcscsere (IKE) egyeztetést indít.
ike cmd=remove	IKE alagutakat deaktivál
ike cmd=list	IKE alagutakat listáz
ikedb	Felületet biztosít az IKE felület adatbázishoz
gentun	Létrehoz egy alagút meghatározást
mktun	Aktivál egy alagút meghatározást
chtun	Módosít egy alagút meghatározást
rmtun	Eltávolít egy alagút meghatározást
lstun	Felsorolja az alagút meghatározásokat
exptun	Exportálja az alagút meghatározásokat
imptun	Importálja az alagút meghatározásokat
genfilt	Létrehoz egy szűrőmeghatározást
mkfilt	Aktivál egy szűrőmeghatározást
mvfilt	Áthelyez egy szűrőszabályt
chfilt	Módosít egy szűrőmeghatározást
rmfilt	Eltávolít egy szűrőmeghatározást
lsfilt	Felsorolja a szűrőmeghatározásokat
expfilt	Exportálja a szűrőmeghatározásokat
impfilt	Importálja a szűrőmeghatározásokat
ipsec_convert	Megjeleníti az IP biztonság állapotát
ipsecstat	Megjeleníti az IP biztonság állapotát

Parancs	Rendeltetés
ipsectrcbuf	Kilistázza az IP biztonsági nyomkövetési puffer tartalmát
unloadipsec	Eltávolít egy titkosítási modult

Metódusok listája:

Az alábbiakban látható a metódusok listája.

defipsec
Definiál egy IPv4 vagy IPv6 IP biztonság példányt

cfgipsec
Beállítja és betölti az **ipsec_v4** vagy **ipsec_v6** szolgáltatást

ucfgipsec
Megszünteti az **ipsec_v4** vagy **ipsec_v6** konfigurációját

IP biztonság átállítása:

Átállíthatja az IKE alagútjait, szűrőit és előzetesen megosztott kulcsait az AIX operációs rendszer korábbi változataiból.

IKE alagutak átállítása:

Az alagutak átállításához tegye a következőket:

1. Futtassa a bos.net.ipsec.keymgt.pre_rm.sh parancsfájlt. A parancsfájl futtatásakor az alábbi fájlok kerülnek létrehozásra a /tmp könyvtárban:
   a. p2proposal.bos.net.ipsec.keymgt
   b. p1proposal.bos.net.ipsec.keymgt
   c. p1policy.bos.net.ipsec.keymgt
   d. p2policy.bos.net.ipsec.keymgt
   e. p1tunnel.bos.net.ipsec.keymgt
   f. p2tunnel.bos.net.ipsec.keymgt

   FIGYELEM: Csak egyszer futtassa ezt a parancsfájlt. Ha frissíti az adatbázist és ismét futtatja a parancsfájlt, akkor elveszti az összes fájlt, és nem fogja tudni visszaállítani azokat. Az alagutak átvétele előtt olvassa el a parancsfájlt a következő részben: “A bos.net.ipsec.keymgt.pre_rm.sh parancsfájl” oldalszám: 266.
2. Mentse a parancsfájl által létrehozott fájlokat és a /tmp/lpplevel fájlt egy külső adathordozóra, például egy CD-re vagy egy lemezre.

Előzetesen megosztott kulcsok átállítása:

Az előzetesen megosztott kulcsformátum frissítéséhez végezze el az alábbi lépéseket.

Az IKE alagút előzetesen megosztott kulcs adatbázisa is sérül az átvétel közben. Az előzetesen megosztott kulcs formátum frissítéséhez az átállított rendszeren tegye a következőket:

1. A következő parancs futtatásával mentse el az **ikedb -g** parancs kimenetét:
   ```
   ikedb -g > out.keys
   ```
2. Szerkessze a out.keys fájlt, hogy lecserélje a FORMAT=ASCII bejegyzést FORMAT=HEX bejegyzésre az előzetesen megosztott kulcsformátum használatához.
3. A következő parancs futtatásával adja meg bemenetként az XML fájlt:
   ```
   ikedb -pF out.keys
   ```

Szűrők átállítása:

A szűrők átállításához végezze el az alábbi lépéseket.

1. A SMIT-ben az alábbi lépések végrehajtásával exportálja a szűrőszabály fájlokat a /tmp könyvtárba:
   a. Futtassa a **smitty ipsec4** parancsot.
   b. Válassza ki a További IP biztonság konfiguráció—> IP biztonság szűrőszabályainak beállítása—>IP biztonság szűrőszabályainak exportálása menüpontot.
   c. Adja meg a /tmp könyvtárnevet.
   d. A Szűrőszabályok beállításnál nyomja le az F4 billentyűt, majd a listában válassza ki az **all** lehetőséget.
   e. Az Enter billentyű lenyomásával mentse el a /tmp/ipsec_fltr_rule.exp fájlban található szűrőszabályokat egy külső adathordozóra.

   Hajtsa végre ezt a folyamatot az összes rendszer esetében, amit átvesz az AIX operációs rendszer korábbi változataiból.
2. Másolja át a parancsfájl által létrehozott hat alagút fájlt, a /tmp/lpplevel fájlt és a /tmp/ipsec_fltr_rule.exp fájlt az áttért rendszer /tmp könyvtárába.
3. A bos.net.ipsec.keymgt.post_i.sh parancsfájl futtatásával töltse fel az adatbázist az alagút konfigurációkkal.
4. Az **ikedb -g** parancs futtatásával ellenőrizze, hogy az alagutak benne vannak-e az adatbázisban.

   Megjegyzés: Ha nem látja az alagút információkat az adatbázisban, akkor futtassa ismét a parancsfájlt, de nevezze át az összes *.loaded fájlt a /tmp könyvtárban az eredeti névre.

Az átállított rendszeren a szűrő adatbázis sérült az átállítás után. A következő hibát fogja kapni, ha az **lsfilt** parancsot futtatja az áttért rendszeren:

```
Az ipv4
alapértelmezett szűrőszabály nem kérhető le
```

A szűrő adatbázis frissítéséhez végezze el az alábbi lépéseket:

1. Cserélje le az /etc/security könyvtár ipsec_filter és ipsec_filter.vc fájlját az újonnan átvett rendszer hibátlan fájljaira. Ha nem rendelkezik ezekkel a fájlokkal, akkor kérje a fájlokat az IBM szerviztől.
2. A SMIT-ben az alábbi lépések végrehajtásával importálja a szűrőszabály fájlokat a /tmp könyvtárba:
   a. Futtassa a **smitty ipsec4** parancsot.
   b. Válassza ki a További IP biztonság konfiguráció—> IP biztonság szűrőszabályainak beállítása—>IP biztonság szűrőszabályainak importálása menüpontot.
   c. Adja meg a /tmp könyvtárnevet.
   d. A **Szűrőszabályok** beállításnál nyomja le az **F4** billentyűt, majd a listában válassza ki az **all** lehetőséget.
   e. A szűrőszabályok ismételt létrehozásához nyomja meg az Entert. A szűrőszabályokat a SMIT vagy az **lsfilt** parancs segítségével listázhatja ki.

A bos.net.ipsec.keymgt.pre_rm.sh parancsfájl:

A bos.net.ipsec.keymgt.pre_rm.sh elmenti az alagút adatbázis tartalmát az AIX operációs rendszert futtató rendszeren.

```
#!/usr/bin/ksh
keymgt_installed=`lslpp -Lqc bos.net.ipsec.keymgt 2>/dev/null | awk -F: '{print $6}' | head -1`

if [ ! "$keymgt_installed" ]
then
  exit 0
fi


#  Adatbázis átmásolása egy mentési könyvtárba arra az esetre, ha a módosítás
#  nem sikerülne
```

```
if [ -d /etc/ipsec/inet/DB ]
then
  cp -R /etc/ipsec/inet/DB /etc/ipsec/inet/DB.sav || exit $?
fi

#  Ne feledje azt a szintet, amelyről az áttérést végzi
VRM=$(LANG=C lslpp -Lqc bos.net.ipsec.keymgt 2>/dev/null | awk -F: '{print $3}' | \
awk -F. '{print $1"."$2"."$3}')
VR=${VRM%.*}
echo $VRM > /tmp/lpplevel

IKEDB=$(which ikedb) || IKEDB=/usr/sbin/ikedb

XMLFILE=/tmp/full_ike_database.bos.net.ipsec.keymgt
PSKXMLFILE=/tmp/psk_ike_database.bos.net.ipsec.keymgt

#  Az ikedb meglétének vizsgálata.
if [ -f $IKEDB ]
then

  # Ha az egyik alábbi ikedb hívás meghiúsul, az nem probléma.  Csak távolítsa el az
  # eredményül kapott fájlt (amely szemetet tartalmazhat) és folytassa a működést.  A post_i
  # parancsfájl egyszerűen nem importálja a fájlt, ha nem létezik, amely
  # az IKE adatbázis egy részének vagy egészének elvesztését eredményezi, de érdemesebb
  # a parancsfájlt hibakóddal kiléptetni, amely a teljes átállás
  # meghiúsulását eredményezi.

  $IKEDB -g > $XMLFILE
  if [ $? -ne 0 ]
  then
    rm -f $XMLFILE  || exit $?
  fi

  if [[ $VR = "5.1" ]]; then
    # Ez egy speciális eset. Az ikedb 5.1 változata az egyetlen, amely nem
    # tartalmazza az előzetesen megosztott kulcsokat a teljes adatbázis
    # kimenetben. Ezért ezeket külön kell visszakeresni.
    $IKEDB -g -t IKEPresharedKey > $PSKXMLFILE
    if [ $? -ne 0 ]
    then
     rm -f $PSKXMLFILE  || exit $?
    fi
  fi

#  Ellenőrizzük, hogy az ikegui parancs telepítve van-e
elif [ -f /usr/sbin/ikegui ]
then

  #  Adatbázis információk visszakeresése és elmentése a /tmp könyvtárba
  /usr/sbin/ikegui 0 1 0 0 > /tmp/p1proposal.bos.net.ipsec.keymgt 2>/dev/null
  RC=$?
  if [[ $RC -ne 0 ]]
  then
    rm -f /tmp/p1proposal.bos.net.ipsec.keymgt || exit $?
  fi

  /usr/sbin/ikegui 0 1 1 0 > /tmp/p1policy.bos.net.ipsec.keymgt 2>/dev/null
  RC=$?
  if [[ $RC -ne 0 ]]
  then
    rm -f /tmp/p1policy.bos.net.ipsec.keymgt || exit $?
  fi

  /usr/sbin/ikegui 0 2 0 0 > /tmp/p2proposal.bos.net.ipsec.keymgt 2>/dev/null
  RC=$?
  if [[ $RC -ne 0 ]]
  then
```

```
    rm -f /tmp/p2proposal.bos.net.ipsec.keymgt || exit $?
  fi

  /usr/sbin/ikegui 0 2 1 0 > /tmp/p2policy.bos.net.ipsec.keymgt 2>/dev/null
  RC=$?
  if [[ $RC -ne 0 ]]
  then
    rm -f /tmp/p2policy.bos.net.ipsec.keymgt || exit $?
  fi

  /usr/sbin/ikegui 0 1 2 0 > /tmp/p1tunnel.bos.net.ipsec.keymgt 2>/dev/null
  RC=$?
  if [[ $RC -ne 0 ]]
  then
    rm -f /tmp/p1tunnel.bos.net.ipsec.keymgt || exit $?
  fi

  /usr/sbin/ikegui 0 2 2 0 > /tmp/p2tunnel.bos.net.ipsec.keymgt 2>/dev/null
  RC=$?
  if [[ $RC -ne 0 ]]
  then
    rm -f /tmp/p2tunnel.bos.net.ipsec.keymgt || exit $?
  fi

fi
```

A bos.net.ipsec.keymgt.post_i.sh parancsfájl:

A bos.net.ipsec.keymgt.post_i.sh parancsfájl betölti az alagút adatbázis tartalmát az AIX operációs rendszert futtató átállított rendszerre.

```
#!/usr/bin/ksh

function PrintDot {
    echo "echo \c"
    echo "\".\c"
    echo "\\\c\c"
    echo "\"\c"
    echo
}

function P1PropRestore {
    while :
    do
        read NAME
        read MODE
        if [[ $? = 0 ]]; then
            echo "ikegui 1 1 0 $NAME $MODE \c"
            MORE=1
            while [[ $MORE = 1 ]];
            do
                read AUTH
                read HASH
                read ENCRYPT
                read GROUP
                read TIME
                read SIZE
                read MORE
                echo "$AUTH $HASH $ENCRYPT $GROUP $TIME $SIZE $MORE \c"
            done
            echo " > /dev/null 2>&1"
            PrintDot
        else
            return 0
        fi
    done
}
```

```
function P2PropRestore {
    while :
    do
        read NAME
        FIRST=yes
        MORE=1
        while [[ $MORE = 1 ]];
        do
            read PROT
            if [[ $? = 0 ]]; then
                read AH_AUTH
                read ESP_ENCR
                read ESP_AUTH
                read ENCAP
                read TIME
                read SIZE
                read MORE
                if [[ $FIRST = "yes" ]]; then
                    echo "ikegui 1 2 0 $NAME $MODE \c"
                fi
                echo "$PROT $AH_AUTH $ESP_ENCR $ESP_AUTH $ENCAP $TIME $SIZE $MORE \c"
                FIRST=no
            else
                return 0
            fi
        done
        echo " > /dev/null 2>&1"
        PrintDot
    done
}

function P1PolRestore {
    while :
    do
        read NAME
        read ROLE
        if [[ $? = 0 ]]; then
            read TIME
            read SIZE
            read OVERLAP
            read TTIME
            read TSIZE
            read MIN
            read MAX
            read PROPOSAL
            echo "ikegui 1 1 1 $NAME $ROLE $OVERLAP $TTIME $TSIZE $MIN $MAX 1 0 0 $PROPOSAL > \
/dev/null 2>&1"
            PrintDot
        else
            return 0
        fi
    done
}

function P2PolRestore {
    while :
    do
        read NAME
        read ROLE
        if [[ $? = 0 ]]; then
            read IPFS
            read RPFS
            read TIME
            read SIZE
            read OVERLAP
            read TTIME
```

```
            read TSIZE
            read MIN
            read MAX
            echo "ikegui 1 2 1 $NAME $ROLE $IPFS $RPFS $OVERLAP $TTIME $TSIZE $MIN $MAX 1 0 0 \c"
            MORE=1
            while [[ $MORE = 1 ]];
            do
                read PROPOSAL
                read MORE
                echo "$PROPOSAL $MORE \c"
                FIRST=no
            done
        else
            return 0
        fi
        echo " > /dev/null 2>&1"
        PrintDot
    done
}

function P1TunRestore {
    while :
    do
        read TUNID
        read NAME
        if [[ $? = 0 ]]; then
            read LID_TYPE
            read LID
            if [[ $LPPLEVEL = "4.3.3" ]]; then
                read LIP
            fi
            read RID_TYPE
            read RID
            read RIP
            read POLICY
            read KEY
            read AUTOSTART
            echo "ikegui 1 1 2 0 $NAME $LID_TYPE \"$LID\" $LIP $RID_TYPE \"$RID\" \
$RIP $POLICY $KEY $AUTOSTART > /dev/null 2>&1"
            PrintDot
        else
            return 0
        fi
    done
}

function P2TunRestore {
    while :
    do
        read TUNID
        read NAME
        if [[ $? = 0 ]]; then
            read P1TUN
            read LTYPE
            read LID
            read LMASK
            read LPROT
            read LPORT
            read RTYPE
            read RID
            read RMASK
            read RPROT
            read RPORT
            read POLICY
            read AUTOSTART
            echo "ikegui 1 2 2 0 $NAME $P1TUN $LTYPE $LID $LMASK $LPROT $LPORT $RTYPE
              \$RID $RMASK $RPROT $RPORT $POLICY $AUTOSTART > /dev/null 2>&1"
```

```
            PrintDot
        else
            return 0
        fi
    done
}

function allRestoreWithIkedb {

    ERRORS=/tmp/ikedb_msgs.bos.net.ipsec.keymgt
    echo > $ERRORS
    $IKEDB -p $XMLFILE 2>> $ERRORS
    if [ -f $PSKXMLFILE ]
    then
        $IKEDB -p $PSKXMLFILE 2>> $ERRORS
    fi

}

P1PROPFILE=/tmp/p1proposal.bos.net.ipsec.keymgt
P2PROPFILE=/tmp/p2proposal.bos.net.ipsec.keymgt
P1POLFILE=/tmp/p1policy.bos.net.ipsec.keymgt
P2POLFILE=/tmp/p2policy.bos.net.ipsec.keymgt
P1TUNFILE=/tmp/p1tunnel.bos.net.ipsec.keymgt
P2TUNFILE=/tmp/p2tunnel.bos.net.ipsec.keymgt
XMLFILE=/tmp/full_ike_database.bos.net.ipsec.keymgt
PSKXMLFILE=/tmp/psk_ike_database.bos.net.ipsec.keymgt
CMD_FILE=/tmp/commands
IKEDB=$(which ikedb) || IKEDB=/usr/sbin/ikedb

echo "building ISAKMP database \n"
$IKEDB -x || exit $?

if [ -f $XMLFILE ]; then
    echo "\nRestoring database entries\c"
    allRestoreWithIkedb
    echo "\ndone\n"

elif [ -f /tmp/*.bos.net.ipsec.keymgt ]; then
    echo "\nRestoring database entries\c"

    LPPLEVEL=`cat /tmp/lpplevel`

    echo > $CMD_FILE
    touch $P1PROPFILE; P1PropRestore < $P1PROPFILE >> $CMD_FILE
    touch $P2PROPFILE; P2PropRestore < $P2PROPFILE >> $CMD_FILE
    touch $P1POLFILE; P1PolRestore < $P1POLFILE >> $CMD_FILE
    touch $P2POLFILE; P2PolRestore < $P2POLFILE >> $CMD_FILE
    touch $P1TUNFILE; P1TunRestore < $P1TUNFILE >> $CMD_FILE
    touch $P2TUNFILE; P2TunRestore < $P2TUNFILE >> $CMD_FILE

    mv $P1PROPFILE ${P1PROPFILE}.loaded
    mv $P2PROPFILE ${P2PROPFILE}.loaded
    mv $P1POLFILE ${P1POLFILE}.loaded
    mv $P2POLFILE ${P2POLFILE}.loaded
    mv $P1TUNFILE ${P1TUNFILE}.loaded
    mv $P2TUNFILE ${P2TUNFILE}.loaded

    ksh $CMD_FILE

    echo "done\n"
fi
```

Hálózati fájlrendszer biztonság

A hálózati fájlrendszer (NFS) egy széles körben elérhető technológia, ami lehetővé teszi adatok megosztását a hálózaton különböző gazdagépek között.

Az NFS a DES-en felül a Kerberos 5 hitelesítést is támogatja. A Kerberos 5 biztonság egy RPCSEC_GSS-nek nevezett protokoll mechanizmus alatt biztosított.

Az alap UNIX hitelesítési rendszerhez képest az NFS értelmezést nyújt a felhasználók és gépek üzenet-üzenet alapú hitelesítéséhez a hálózaton. A járulékos hitelesítési rendszer az Adattitkosítási Szabvány (DES) titkosítást és nyilvános kulcsú titkosítást használja.

Az NFS a DES-en felül a Kerberos 5 hitelesítést is támogatja. A Kerberos 5 biztonság egy RPCSEC_GSS-nek nevezett protokoll mechanizmus alatt biztosított. A Kerberos hitelesítés NFS-en történő adminisztrálásának és használatának leírásához tekintse meg a következő kézikönyvet: *NFS Adminisztrációs kézikönyv*.

Hálózati fájlrendszer biztonságossá tételének általános irányelvei

Számos irányelv segít a Hálózati fájlrendszer (NFS) biztonságossá tételében.

- Győződjön meg róla, hogy a legfrissebb szoftverjavítások telepítve vannak. A biztonsági kérdéseket érintő javítások különösen fontosak. Az adott infrastruktúra minden szoftverét karban kell tartani. Ha például az operációs rendszer javításait telepíti de a webszerverét nem, akkor egy támadónak olyan támadási felületet ad a környezethez, amelyet elkerülhet, ha a webszervert is frissíti. Ha elő szeretne fizetni a legfrissebb biztonsági információkat tartalmazó IBM System p Security Alerts kiadványra, akkor látogasson el a következő webcímre: http://www14.software.ibm.com/webapp/set2/subscriptions/pqvcmjd.
- Úgy állítsa be az NFS szervert, hogy a szerver a lehető legkevesebb jogosultsággal exportálja a fájlrendszereket. Ha a felhasználóknak csak a fájlrendszer olvasásra van szükségük, akkor a felhasználók nem rendelkezhetnek írási jogosultsággal a fájlrendszerhez. Így a felhasználók még véletlenül sem írhatják felül a fontos adatokat, nem módosíthatják a konfigurációs fájlokat és nem írhatják felül a végrehajtható kódokat az exportált fájlrendszerben. A jogosultságokat a SMIT segítségével vagy az /etc/exports fájl közvetlen szerkesztésével adhatja meg.
- Úgy állítsa be az NFS szervert, hogy a szerver kifejezetten azoknak a felhasználóknak exportálja a fájlrendszereket, akiknek erre szükségük van. A legtöbb NFS megvalósításban megadhatja, hogy mely NFS klienseknek lehet hozzáférése egy adott fájlrendszerhez. Így elkerülheti hogy jogosulatlan felhasználók férjenek hozzá a fájlrendszerekhez. Különösen ne állítsa be az NFS szerver a fájlrendszer saját magába való exportálására.
- Az exportált fájlrendszereknek a saját partíciójukban kell lenniük. Egy támadó károsíthatja a rendszert, ha addig ír egy exportált fájlrendszerbe, amíg a fájlrendszer meg nem telik. Így a fájlrendszer elérhetetlenné válhat a többi alkalmazás és felhasználó számára.
- Ne engedélyezze az NFS klienseknek, hogy root vagy ismeretlen felhasználói azonosítókkal férjenek hozzá a fájlrendszerhez. A legtöbb NFS megvalósításban meg lehet adni a jogosult vagy ismeretlen felhasználóktól érkező kérések leképezését egy jogosulatlan felhasználóra. Így elkerülheti az olyan helyzeteket, amikor a támadó jogosult felhasználóként próbálja meg elérni a fájlokat vagy fájlműveleteket végrehajtani.
- Ne engedélyezze az NFS kliensek számára a suid és sgid programok futtatását az exportált fájlrendszereken. Így az NFS felhasználók nem futtathatnak gyanús kódokat a jogosultságokkal. Ha a támadó képes elérni, hogy a végrehajtható fájl egy jogosult felhasználó vagy csoport tulajdonában legyen, akkor komoly károkat okozhat az NFS szerveren. Ezt az **mknfsmnt -y** parancs kapcsolóval adhatja meg.
- Használjon Biztonságos NFS-t. A Biztonságos NFS DES titkosítást használ az RPC tranzakciókban résztvevő hosztok hitelesítéséhez. Az RPC egy protokoll, amelyet az NFS a kérések hosztok közötti kommunikációjához használ. A Biztonságos NFS az RPC kérések időpecsétjének titkosításával kiküszöböli, hogy egy támadó az RPC kéréseket meghamisítsa. Ha a fogadó sikeresen dekódolja az időpecsétet és igazolja a helyességét, akkor ez megerősíti, hogy az RPC kérés egy megbízható hosztról érkezett.
- Ha az NFS-re nincs szükség, akkor kapcsolja ki. Így csökkentheti a támadási felületet a behatolókkal szemben.

Az NFS az AES titkosítási típus használatát is támogatja a Kerberos 5 hitelesítéshez a Triple DES és Single DES titkosításon felül. A Kerberos 5 AES titkosítási típus használatára való beállításához tekintse meg az NFS rendszerkezelési útmutatót.

Kapcsolódó fogalmak:

“Hálózati fájlrendszer biztonság” oldalszám: 271

Kapcsolódó tájékoztatás:

Ellenőrzőlista az NFS konfigurálásához
NFS démonok elindítása rendszerindításkor
NFS szerver konfigurálása
NFS kliens konfigurálása
Azonosságleképezés
NFS fájlrendszer exportálása
Hálózat beállítása RPCSEC-GSS számára
NFS fájlrendszer exportálásának megszüntetése
Exportált fájlrendszer módosítása
Root felhasználói hozzáférés exportált fájlrendszerhez
NFS fájlrendszer explicit felépítése
Alrendszer automatikus felépítése
Előre meghatározott NFS felépítések létesítése
Előre meghatározott NFS felépítések eltávolítása
NFS exportfájlok
mknfsmnt parancs

Hálózati fájlrendszer hitelesítés

Az NFS a DES algoritmust különböző célokra használja. Az NFS DES algoritmussal titkosítja az időpecséteket az NFS szerverek és a kliensek között küldött Távoli eljáráshívási (RPC) üzenetekben. Ez a titkosított időpecsét hitelesíti a számítógépet, ugyanúgy mint ahogyan a token hitelesíti a küldőt.

Mivel az NFS az NFS kliensek és szerverek közötti minden egyes RPC üzenetet képes hitelesíteni, így ez egy további, nem kötelező biztonsági szintet jelent minden egyes rendszernél. A fájlrendszerek alapértelmezésben szabványos UNIX hitelesítéssel kerülnek exportálásra. Ha ki szeretné használni ezt a további biztonsági szintet, akkor a fájlrendszer exportálásakor adja meg a secure beállítást.

Nyilvános kulcsú titkosítás biztonságos Hálózati fájlrendszerhez:

A felhasználó nyilvános és a titkos kulcsa is a publickey.byname adatbázisban került hálózatnév alapján eltárolásra és indexelésre.

A titkos kulcs a bejelentkezési jelszóval van DES titkosítva. A **keylogin** parancs a bejelentkezési jelszóval dekódolja a titkosított titkos kulcsot, majd átadja a védett helyi kulcsszervernek, amely elmenti a jövőbeni RPC tranzakciókhoz. A felhasználók nem ismerik a saját nyilvános és titkos kulcsaikat, mivel az **yppasswd** parancs a bejelentkezési jelszó módosításán túl automatikusan hozza létre a nyilvános és titkos kulcsokat.

A keyserv démon az egyes NIS számítógépeken futó RPC szolgáltatás. A NIS-en belül a **keyserv** az alábbi nyilvános kulcs szubrutinokat futtatja:

- **key_setsecret** szubrutin
- **key_encryptsession** szubrutin
- **key_decryptsession** szubrutin

A **key_setsecret** szubrutin mondja meg a kulcsszervernek, hogy tárolja el a felhasználó titkos kulcsát (SK_A) a jövőbeni használatra. Általában a **keylogin** parancs hívja meg. A kliensprogram a **key_encryptsession** szubrutin meghívásával előállítja a titkosított párbeszédkulcsot, amely az első RPC tranzakcióban kerül átadásra a szervernek. A kulcsszerver a közös kulcs létrehozásához megkeresi a szerver nyilvános kulcsát, és összepárosítja a kliens titkos kulcsával (amelyet az előző **key_setsecret** szubrutin állított be). A szerver kéri a kulcsszervert, hogy dekódolja a párbeszéd kulcsot a **key_decryptsession** szubrutin meghívásával.

A hívó neve egyértelmű ezekben a szubrutin hívásokban, de a nevet valahogy hitelesíteni kell. A kulcsszerver nem használhat DES hitelesítést ehhez, mivel az holtpontot eredményezne. A kulcsszerver úgy oldja meg ezt a problémát,

hogy a titkos kulcsokat felhasználói azonosítónként (UID) tárolja, és csak a helyi root folyamatoknak ad kéréseket. A kliensfolyamat futtatja a root felhasználó **setuid** szubrutinját, amely a kliens helyett kiadja a kérést, és megadja a kulcsszervernek a kliens tényleges UID-jét.

Hálózati fájlrendszer hitelesítés követelményei:

Az NFS hitelesítés azon alapul, hogy a küldő képes titkosítani a pontos időt, és a címzett képes dekódolni azt, és összehasonlítani a saját idejével.

A folyamat az alábbi követelményeket támasztja:

- A két ügynöknek meg kell egyeznie a pontos időben.
- A küldőnek és a címzettnek ugyanazt a DES titkosítási kulcsot kell használnia.

Pontos idő egyeztetése:

Ha a hálózat idő szinkronizálást használ, akkor a timed démon tartja szinkronban a kliens és a szerver idejét. Ellenkező esetben a kliens a szerveróra alapján számítja ki a megfelelő időpecsétet.

Ehhez a kliens még az RPC szekció indítása előtt meghatározza a szerveridőt, és kiszámítja a különbséget a saját ideje és a szerveridő között. A kliens ennek megfelelően állítja be az időpecsétet. Ha az RPC szekció közben a szerveridő kiesik a szinkronból, és elkezdi visszautasítani a kliens kéréseit, akkor a kliens ismét meghatározza a szerveridőt.

Azonos DES kulcs használata:

A kliens és a szerver nyilvános kulcs kriptográfiával határozza meg ugyanazt a DES titkosítást.

Tetszőleges A klienshez és B szerverhez létezik egy olyan *közös kulcs*, amelyet csak A és B tud levezetni. A kliens a következő formula kiszámításával származtatja a közös kulcsot:

$$K_{AB} = PK_B^{SK}A$$

ahol *K* a közös kulcs, a *PK* a nyilvános kulcs az *SK* pedig a titkos kulcs, és minden kulcs 128 bites szám. A szerver a következő formula kiszámításával származtatja a közös kulcsot:

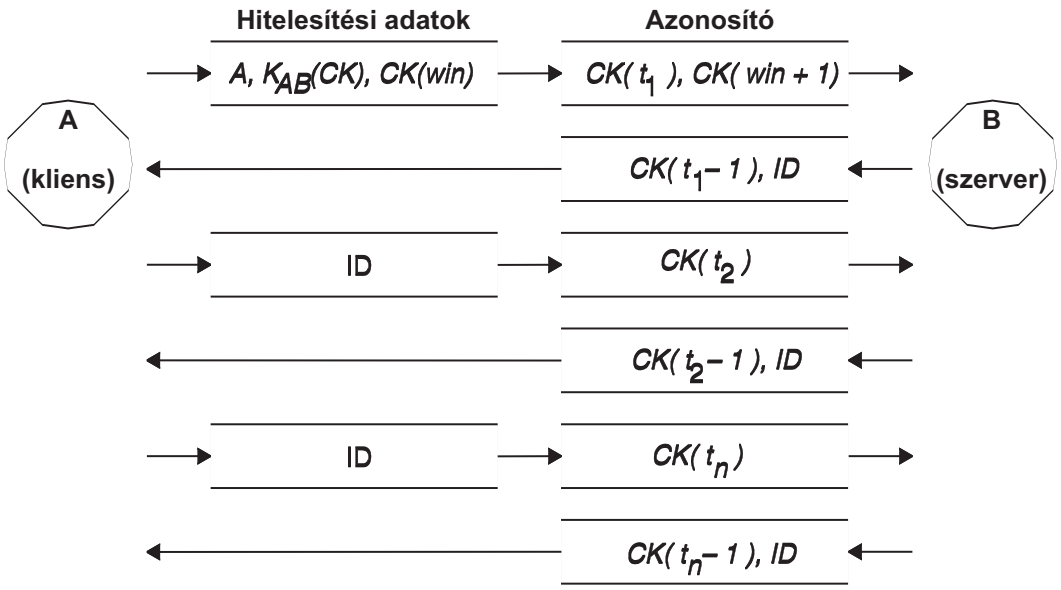
$$K_{AB} = PK_A^{SK}B$$

Csak a szerver és a kliens tudja kiszámítani ezt a közös kulcsot, mivel a számításhoz ismerni kell egy titkos kulcsot és egy másikat. Mivel a közös kulcs 128 bites és a DES 56 bites kulcsot használ, ezért a kliens és a szerver a közös kulcsból egy 56 bites DES kulcsot képez.

Hálózati fájlrendszer hitelesítési folyamat:

Ha egy kliens kommunikálni szeretne egy szerverrel, akkor véletlenszerűen létrehoz egy kulcsot, amellyel titkosítja az időpecsétet. Ezt a kulcsot *párbeszéd kulcsnak* (*CK*) nevezik.

A kliens a közös DES kulccsal titkosítja a párbeszéd kulcsot (lásd: Hitelesítés követelményei), és elküldi a szerverhez az első RPC tranzakcióban. A folyamatot a következő ábra szemlélteti.



15. ábra: Hitelesítés folyamata. Ez az ábra mutatja be a hitelesítési folyamatot.

Az ábrán látható, hogy A kliens csatlakozik a B szerverhez. A *K*(*CK*) azt jeleneti, hogy a *CK* a közös *K* DES kulccsal van titkosítva. Az első kérésben a kliens RPC hitelesítési adatai tartalmazzák a kliens nevét (*A*), a párbeszéd kulcsot (*CK*) és a *win* (window) nevű *CK*-val titkosított változót. (Az alapértelmezett ablakméret a 30 perc.) A kliens azonosító az első kérésben tartalmazza a titkosított időpecsétet és a megadott ablak egy titkosított azonosítóját - *win* + 1. Az ablak azonosító nehezebben találja ki a megfelelő azonosítót, ami növeli a biztonságot.

A kliens hitelesítése után a szerver eltárolja a következő elemeket egy azonosító táblában:

- Kliens neve, *A*
- Párbeszéd kulcs, *CK*
- Ablak
- Időbélyeg

A szerver csak olyan időpecséteket fogad el, amelyek kronológiailag az utoljára látott időpecsét után következnek, így a megválaszolt tranzakciókat biztosan visszautasítja. A szerver az azonosítóban egy index azonosítót ír a meghatalmazás táblába a kliensen, valamint a kliens időpecsétje - 1 értéket a *CK* kulccsal titkosítva. A kliens tudja, hogy csak a szerver küldhetett ilyen azonosítót, mivel csak a szerver ismeri az időpecsétet, amelyet a kliens küldött. Azért kell egyet kivonni az időpecsétből, hogy biztosan ne legyen érvényes, és ne lehessen ismét felhasználni kliens azonosítóként. Az első RPC tranzakció után a kliens csak a saját azonosítóját és egy titkosított időpecsétet küld a szerverre, a szerver pedig az időpecsét - 1 értéket küldi vissza a *CK* kulccsal titkosítva.

Hálózati entitások elnevezése a DES hitelesítéshez

A DES hitelesítés a hálózati nevek használatával végzi a saját elnevezését.

A *hálózati név* egy nyomtatható karaktersorozat a hitelesítéshez. A rendszer a nyilvános és titkos kulcsokat nem felhasználói név alapján, hanem hálózati név alapján tárolja. A netid.byname NIS adatbázis képezi le a hálózati neveket helyi UID-re és csoport hozzáférési listára.

A felhasználói nevek egyediek a tartományokon belül. A hálózati nevek az operációs rendszer és a felhasználói azonosító NIS és Internet tartománynevekkel való összefűzésével jönnek létre. A tartományok elnevezésének egy jó megállapodása az Internet tartománynév (com, edu, gov, mil) hozzáfűzése a helyi tartománynévhez.

Hálózati nevek vannak hozzárendelve a számítógépekhez és a felhasználókhoz is. A számítógépek hálózati nevét ugyanúgy kell meghatározni, mint a felhasználókét. A hal nevű gép hálózati neve például az eng.xyz.com tartományban unix.hal@eng.xyz.com. A megfelelő hitelesítés fontos a lemeznélküli gépek számára, amelyek teljes hozzáférést igényelnek a saját könyvtáraikhoz a hálózaton.

Ha a felhasználókat bármely távoli tartományból hitelesíteni szeretné, akkor hozzon létre bejegyzéseket a számukra két NIS adatbázisban. Az egyik egy bejegyzés a nyilvános és titkos kulcsuk számára. A másik a helyi UID és csoport hozzáférési lista leképezéseket tartalmazza. A távoli tartományok felhasználói minden helyi hálózati szolgáltatáshoz hozzáférnek, ugyanúgy mint az NFS és a távoli bejelentkezéseknél.

A /etc/publickey fájl

Az /etc/publickey fájl neveket és nyilvános kulcsokat tartalmaz, amelyeket a NIS használ a publickey leképezés létrehozásához.

A publickey leképezést biztonságos hálózatkezeléshez használják. A fájl minden bejegyzése egy hálózati felhasználói névből (amely egy felhasználóra vagy egy hosztnévre hivatkozik), felhasználói nyilvános kulcsból (egy hexadecimális jelölésben), egy kettőspontból, és a felhasználó titkosított titkos kulcsából (szintén hexadecimális jelölésben) áll. Alapértelmezésben az /etc/publickey fájlban csak egy felhasználó van, a nobody.

Ne használjon szövegszerkesztőt az /etc/publickey fájl módosításához, mivel a fájl titkosítási kulcsokat tartalmaz. A /etc/publickey fájl módosításához használja a **chkey** vagy a **newkey** parancsot.

Nyilvános kulcskezelési rendszerek rendszerbetöltési szempontjai

A számítógép áramkimaradás utáni újraindításakor minden eltárolt titkos kulcs elveszik, és semmilyen folyamat nem tud hozzáférni a biztonságos hálózati szolgáltatásokhoz (pl.: egy NFS felépítése). A root folyamatok tovább futhatnak, ha valaki megadja a jelszót, amely dekódolja a root felhasználó titkos kulcsát. A megoldás a root felhasználó dekódolt titkos jelszavának eltárolása egy olyan fájlban, amelyet a kulcsszerver képes olvasni.

Nem minden **setuid** szubrutin működik megfelelően. Ha egy **setuid** szubrutint az A tulajdonos indított el, de az A felhasználó a számítógép elindítása óta nem lépett be, akkor a szubrutinnak nincs hozzáférése a biztonságos hálózati szolgáltatásokhoz A felhasználóként. A legtöbb **setuid** szubrutin hívás tulajdonosa mindenesetre a root felhasználó, és a rendszer a root felhasználó titkos kulcsát indításkor mindig eltárolja.

Biztonságos hálózati fájlrendszer teljesítmény szempontok

Az NFS számos módon hat a rendszer teljesítményére.

- A kliensnek és a szervernek is ki kell számolnia a közös kulcsot. A közös kulcs kiszámításának ideje körülbelül egy másodperc. Ennek eredményeként körülbelül két másodpercbe telik egy kezdeti RPC kapcsolat létrehozása, mivel a műveletet a kliensnek és a szervernek is végre kell hajtania. A kezdeti RPC kapcsolat létrehozása után a kulcsszerver ideiglenesen eltárolja az előző számítások eredményét, így nem kell mindig kiszámítani a közös kulcsot.
- Minden RPC tranzakció a következő DES titkosítási műveleteket igényli:
  1. A kliens titkosítja a kérés időpecsétjét.
  2. A szerver dekódolja azt.
  3. A szerver titkosítja a válasz időpecsétjét.
  4. A kliens dekódolja azt.

Mivel a biztonságos NFS csökkentheti a rendszer teljesítményét, ezért mérlegelje a megnövelt biztonság előnyeit a rendszer teljesítmény követelményeivel szemben.

Biztonságos hálózati fájlrendszer ellenőrzőlista

Ezzel az ellenőrzőlistával győződhet meg róla, hogy az NFS megfelelően működik.

- Ha egy fájlrendszert a **-secure** kapcsolóval építi fel egy kliensen, akkor a szerver nevének meg kell egyeznie a szerver hosztnevével az /etc/hosts fájlban. Ha a hosztnév feloldásához névszervert használ, akkor győződjön meg róla, hogy a névszerver által visszaadott hoszt információk megfelelnek az /etc/hosts fájl bejegyzésének

információival. Ha ezek a nevek nem egyeznek, akkor hitelesítési hibák fordulhatnak elő, mivel a gépek hálózati neve az /etc/hosts fájl elsődleges bejegyzésén alapulnak, a **publickey** adatbázis kulcsait pedig a rendszer a hálózati név alapján éri el.

- Ne használjon biztonságos és nem biztonságos exportokat és felépítéseket vegyesen. Ellenkező esetben elképzelhető, hogy a rendszer a fájlhozzáféréseket nem megfelelően határozza meg. Ha például egy kliens a **-secure** kapcsoló nélkül épít fel egy biztonságos fájlrendszert, vagy a **-secure** kapcsolóval épít fel egy nem biztonságos rendszert, akkor a felhasználók nobody felhasználóként kapnak hozzáférést, nem a saját azonosítójuk alapján. Ez akkor is előfordul, ha a NIS számára ismeretlen felhasználó fájlokat próbál meg létrehozni vagy módosítani egy biztonságos fájlrendszeren.
- Mivel a NIS-nek terjesztenie kell az új adatbázist a **chkey** és a **newkey** parancsok minden egyes használata után, ezért csak akkor használja ezeket a parancsokat, ha kicsi a hálózati terhelés.
- Ne törölje az /etc/keystore vagy az /etc/.rootkey fájlt. A számítógépek újratelepítésekor, áthelyezésekor vagy frissítésekor mindig mentse el az /etc/keystore és az /etc/.rootkey fájlokat.
- Utasítsa a felhasználókat, hogy a jelszavak módosításához a **passwd** parancs helyett használják az **yppasswd** parancsot. Így a jelszavak és a magánkulcsok szinkronban maradnak.
- Mivel a **login** parancs nem keresi ki a publickey adatbázisból a kulcsokat a **keyserv** démon számára, ezért a felhasználónak a **keylogin** parancsot kell végrehajtania. A **keylogin** parancsot beírhatja minden egyes felhasználó profile fájljába, hogy a parancs minden bejelentkezéskor automatikusan futtatásra kerüljön. A **keylogin** parancs megköveteli, hogy a felhasználó ismét megadja jelszavát.
- Ha kulcsokat generál a root felhasználó számára az egyes hosztokon a **newkey -h** vagy a **chkey** paranccsal, akkor a **keylogin** parancs futtatásával adja át az új kulcsokat a **keyserv** démonnak. A rendszer a kulcsokat az /etc/.rootkey fájlban tárolja, amelyet a **keyserv** démon minden egyes elinduláskor kiolvas.
- Időnként ellenőrizze, hogy az **yppasswdd** és **ypupdated** démonok futnak-e a NIS főszerveren. Ezekre a démonokra szükség van a publickey adatbázis karbantartásához.
- Időnként ellenőrizze, hogy a **keyserv** démon fut-e minden biztonságos NFS-t használó számítógépen.

Biztonságos hálózati fájlrendszer beállítása

NIS elsődleges és másodlagos szervereken biztonságos NFS konfigurálásához tegye a következőket.

1. A NIS főszerveren hozzon létre egy bejegyzést a NIS /etc/publickey fájl összes felhasználójához a **newkey** paranccsal az alábbiak szerint:
   - Szabványos felhasználó esetén írja be a következő parancsot:

     ```
     smit newkey
     ```

     VAGY

     ```
     newkey -u felhasználónév
     ```

     Root felhasználó esetén a hoszt számítógépen írja be a következő parancsot:

     ```
     newkey -h hosztnév
     ```

   - A felhasználók létrehozhatják a saját nyilvános kulcsukat a **chkey** vagy a **newkey** parancs segítségével.
2. Hozza létre a NIS publickey leképezést. A megfelelő NIS publickey.byname leképezés csak NIS szervereken található meg.
3. Tegye aktívvá a következő szakaszokat az /etc/rc.nfs fájlban:

   ```
   #if [ -x /usr/sbin/keyserv ]; then
   #  startsrc -s keyserv
   #fi
   #if [ -x /usr/lib/netsvc/yp/rpc.ypupdated -a -d /etc/yp/`domainname` ]; then
   #  startsrc -s ypupdated
   #fi
   #DIR=/etc/passwd
   #if [ -x /usr/lib/netsvc/yp/rpc.yppasswdd -a -f $DIR/passwd ]; then
   #  startsrc -s yppasswdd
   #fi
   ```

4. Indítsa el a **keyserv**, **ypupdated** és **yppasswdd** démonokat a **startsrc** parancs segítségével.

Az biztonságos NFS beállításához a NIS klienseken indítsa el a **keyserv** démont a **startsrc** paranccsal.

Fájlrendszer exportálása biztonságos hálózati fájlrendszer segítségével

Biztonságos hálózati fájlrendszert (NFS) a következő eljárások valamelyikével exportálhat.

- Biztonságos NFS fájlrendszer SMIT segítségével exportálásához tegye a következőket:
  1. Az **lssrc -g nfs** parancs kiadásával győződjön meg róla, hogy az NFS már fut. A kimenetnek azt kell jeleznie, hogy az nfsd és az rpc.mountd démon aktív.
  2. Győződjön meg róla, hogy a publickey adatbázis létezik, illetve hogy a keyserv démon fut. További információk: “Biztonságos hálózati fájlrendszer beállítása” oldalszám: 277.
  3. Futtassa a **smit mknfsexp** gyorselérést.
  4. Adja meg az Exportálandó könyvtár elérési útja, a Könyvtár exportálásának módja, a Könyvtár exportálása most, rendszer újraindítása vagy mindkettő beállításokat. A Biztonságos beállítás használata mezőben adja meg az Igen beállítást.
  5. Adja meg a további nem kötelező jellemzőket, vagy fogadja el az alapértelmezett értékeket.
  6. Lépjen ki a SMIT-ből. Ha az /etc/exports fájl nem létezik, akkor a rendszer létrehozza.
  7. Ismételje meg a 3 - 6. lépést minden egyes exportálandó könyvtárra.
- Biztonságos NFS fájlrendszer szövegszerkesztővel exportálásához tegye a következőket:
  1. Nyissa meg az /etc/exports fájlt a kedvenc szövegszerkesztőjével.
  2. Hozzon létre bejegyzést minden egyes exportálandó könyvtár számára. Használja a könyvtár teljes elérési útját. Listázza ki az összes exportálandó könyvtárat bal margótól kezdődően. Egyik könyvtár sem tartalmazhat olyan egyéb könyvtárat, amely már ki van exportálva. Az /etc/exports fájl bejegyzéseinek teljes szintaxisát az /etc/exports fájl dokumentációjában találja. A dokumentációból megtudhatja, hogyan kell megadni a secure kapcsolót.
  3. Mentse el és zárja be az /etc/exports fájlt.
  4. Ha az NFS fut, akkor írja be a következő parancsot:

     ```
     /usr/sbin/exportfs -a
     ```

     A **-a** kapcsoló hatására az **exportfs** parancs elküldi az /etc/exports fájlban található információkat a kernelnek.
- Az NFS fájlrendszer ideiglenes exportálásához (az /etc/exports fájl módosítása nélkül) írja be a következő parancsot:

  ```
  exportfs -i -o secure /könyvtárnév
  ```

  A parancsban a könyvtárnév az exportálandó fájlrendszer neve. Az **exportfs -i** parancs megadja a rendszernek, hogy az /etc/exports fájl ne kerüljön ellenőrizésre a megadott könyvtárnál, és hogy minden beállítás közvetlenül a parancssorban kerül megadásra.

Fájlrendszer felkapcsolása Biztonságos hálózati fájlrendszer segítségével

A védett NFS könyvtárakat kifejezett módon felkapcsolhatja

Biztonságos NFS könyvtár kifejezett felkapcsolásához tegye a következőket:

1. A következő paranccsal győződjön meg róla, hogy az NFS szerver kiexportálta a könyvtárat:

   ```
   showmount -e Szervernév
   ```

   A parancsban a Szervernév az NFS szerver neve. A parancs megjeleníti az NFS szerverről kiexportált könyvtárak neveit. Ha a felépítendő könyvtár nem szerepel a listában, akkor exportálja ki a könyvtárat a szerverről.
2. Az **mkdir** parancs segítségével hozzon létre egy helyi felkapcsolási pontot. Az NFS sikeres felépítéséhez léteznie kell az NFS felépítés egy felépítési pont könyvtárnak (vagy helyfoglalónak). Ennek a könyvtárnak üresnek kell lennie. Ezt a felépítési pontot ugyanúgy kell létrehozni, mint a többi könyvtárat, semmilyen különleges attribútumra nincs szükség.
3. Győződjön meg róla, hogy a publickey adatbázis létezik, illetve hogy a keyserv démon fut. További információk: “Biztonságos hálózati fájlrendszer beállítása” oldalszám: 277.
4. Írja be a következő parancsot:

```
mount -o secure szervernév:/remote/directory /local/directory
```

ahol a szervernév az NFS szerver neve, a /remote/directory az NFS szerver felkapcsolandó könyvtára, a /local/directory pedig az NFS kliens felkapcsolási pontja.

Megjegyzés: Biztonságos NFS-t csak a root felhasználó kapcsolhat fel.

Vállalati azonosság leképezés

A mai hálózati környezetek rendszerek és alkalmazások összetett csoportjából állnak, ami több felhasználói nyilvántartás kezelését teszi szükségessé. A több felhasználói nyilvántartás kezelése hamar nagy adminisztrációs problémává válik, ami érinti a felhasználókat, az adminisztrátorokat és az alkalmazásfejlesztőket. A Vállalati azonosság leképezés (EIM) lehetővé teszi az adminisztrátorok és alkalmazásfejlesztők számára, hogy könnyen megoldják ezt a problémát.

Ez a fejezet leírja a problémákat, nagyvonalakban bemutatja a rendelkezésre álló megoldásokat és bemutatja az EIM megközelítést.

Több felhasználói nyilvántartás kezelése

Sok adminisztrátor kezel különböző rendszereket és szervereket tartalmazó hálózatokat, ahol az egyes rendszerek külön, saját módjukon kezelik a felhasználókat, különböző felhasználói nyilvántartásokkal.

Ezekben az összetett rendszerekben az adminisztrátorok minden egyes felhasználói azonosítóért és jelszóért felelősek az összes rendszeren. Az adminisztrátoroknak sokszor szinkronizálniuk kell ezeket az azonosítókat és jelszavakat. A felhasználóknak kényelmetlen több azonosítót és jelszót is észben tartani, és ezeket folyton szinkronizálni. A felhasználói és adminisztrátori többletterhelés igen drága ezekben a hálózatokban, és az adminisztrátorok a teljes hálózat kezelése helyett gyakran értékes időt töltenek a hibás bejelentkezési kísérletek hibaelhárításával illetve az elfelejtett jelszavak alaphelyzetbe állításával.

A több felhasználói nyilvántartás kezelési az alkalmazásfejlesztőknek is problémát okoz, hiszen ők többrétegű, heterogén alkalmazásokat szeretnének fejleszteni. Az ügyfelek több különböző típusú rendszeren rendelkeznek fontos üzleti adatokkal, és minden egyes rendszer saját felhasználói nyilvántartást vezet. Ebből kifolyólag a fejlesztőknek saját felhasználói nyilvántartást és biztonsági szemantikát kell létrehozniuk az alkalmazásaikhoz. Bár ez megoldja az alkalmazásfejlesztők problémáját, de további terhelést jelent a felhasználók és az adminisztrátorok számára.

Vállalati azonosság leképezés aktuális megközelítései

Számos megközelítés próbálja meg megoldani a több felhasználói nyilvántartás kezelésének problémáját, de mindegyik csak hiányos megoldás képes nyújtani. Az Egyszerűsített címtárhozzáférési protokoll (LDAP) például egy osztott felhasználói nyilvántartást biztosít. Ugyanakkor az LDAP-hoz hasonló megoldások használatához az adminisztrátoroknak még egy felhasználói nyilvántartást és biztonsági szemantikát kell kezelniük, és úgy kell módosítaniuk a meglévő alkalmazásaikat, hogy ezeket a nyilvántartásokat használják.

Ezzel a megoldással az adminisztrátoroknak több biztonsági mechanizmust kell kezelniük minden egyes egyedi erőforrásnál, ami növeli az adminisztrációs terhelést, és potenciálisan a biztonsági problémák valószínűségét. Ha több mechanizmus is támogat egy erőforrást, akkor fennáll annak a lehetősége, hogy az egyik mechanizmus hozzáférési jogosultságait módosítjuk, a többi mechanizmusét viszont elfelejtjük. Ilyen biztonsági rés lehet például, ha egy felhasználó hozzáférése kifejezetten le van tiltva egy adott illesztőn, de egy másik illesztőn keresztül van hozzáférése.

A feladat elvégzése után az adminisztrátorok azt veszik észre, hogy nem oldották meg teljesen a problémát. A cégek általában túl sok pénzt fektettek a felhasználói nyilvántartásba és a hozzá tartozó biztonsági szemantikába ahhoz, hogy az ilyen típusú megoldás praktikus legyen a számukra. Új felhasználói nyilvántartás és biztonsági szemantika létrehozása megoldja az alkalmazás szolgáltatók problémáját, de a felhasználók és adminisztrátorok problémáit nem.

Egy másik lehetőség az egyetlen bejelentkezés alkalmazása. Számos termék képes biztosítani a rendszergazdák számára, hogy az összes felhasználói azonosítót és jelszót tartalmazó fájlokat kezeljenek. Ennek a megközelítésnek számos gyengesége van:

- A felhasználóknak csak egyetlen problémáját oldja meg. Habár lehetővé teszi a felhasználók számára, hogy egyetlen azonosítóval és jelszóval több rendszerre bejelentkezzenek, de a felhasználóknak még mindig jelszavakkal kell rendelkezniük más rendszereken, vagy ezeket a jelszavakat kezelni kell.
- Ez a megoldás újabb biztonsági problémát vet fel, mivel ezeket a sima- szöveg vagy titkosított jelszavakat a rendszer ezekben a fájlokban tárolja. A jelszavakat nem szabad sima szöveg fájlokban illetve olyan helyen tárolni, ahol bárki - a adminisztrátor is - könnyen hozzáférhet.
- Ez nem oldja meg az egyéb alkalmazásfejlesztők problémáját, akik heterogén, többrétegű alkalmazásokat fejlesztenek. Nekik továbbra is kiegészítő felhasználói nyilvántartásokat kell használniuk a saját alkalmazásaikhoz.

A gyengeségek ellenére néhány cég használja ezeket a megoldásokat, mivel a megoldások egy kicsit enyhítik a több felhasználói nyilvántartás karbantartásának problémáját.

Vállalati azonosság leképezés használata

Az EIM leírja a kapcsolatokat az felhasználók és bejegyzések (például fájlszerverek és nyomtatószerverek) közötti kapcsolatokat egy cégen belül, valamint a leírja a felhasználókat a cégen belül. Ezenkívül az EIM rendelkezik egy API készlettel, amely lehetővé teszi az alkalmazások számára, hogy kérdéseket tegyenek fel ezekről a kapcsolatokról.

Ha például adott egy felhasználó azonossága egy felhasználói nyilvántartásban, akkor meghatározhatja, hogy egy másik nyilvántartásban melyik azonosság jelenti ugyanazt a személyt. Ha a felhasználót a rendszer hitelesítette egy adott azonossággal, amelyet le lehet képezni egy másik felhasználó nyilvántartás megfelelő azonosságára, akkor a felhasználónak a hitelesítéshez nem kell ismét megadnia a felhasználói azonosítóját. Csak annyit kell tudni, hogy melyik azonosság jelenti ugyanazt a felhasználót a másik felhasználói nyilvántartásban. Így az EIM egy általánosított azonosság leképezési funkciót biztosít a vállalat számára.

A felhasználói azonosítók különböző nyilvántartások közötti leképezése számos előnnyel jár. Először is az alkalmazások használhatnak egy adott nyilvántartást a hitelesítésekhez, és egy teljesen különböző nyilvántartást a felhatalmazásokhoz. Az adminisztrátor például leképezhet egy SAP azonosságot a SAP erőforrásokhoz való hozzáféréshez.

Az azonosság leképezéshez az adminisztrátornak az alábbiakat kell tennie:

1. Hozzon létre olyan EIM azonosítókat, amelyek a vállalat embereit és bejegyzéseit képviselik.
2. Hozzon létre olyan EIM nyilvántartás meghatározásokat, amelyek leírják a vállalatnál már meglévő felhasználói nyilvántartásokat.
3. Határozza meg a kapcsolatot a felhasználói nyilvántartások felhasználói azonosságai és a létrehozott EIM azonosítók között.

A meglévő nyilvántartások kódját nem kell módosítani. Nem kell a felhasználó nyilvántartások összes azonosságát leképezni. Az EIM lehetővé teszi az egy-a-többhöz leképezéseket (más szavakkal egyetlen felhasználó több azonossággal egyetlen felhasználói nyilvántartásban). Az EIM a több-az-egyhez leképezéseket is lehetővé teszi (más szavakkal több felhasználó osztja meg ugyanazt az azonosságot egy felhasználói nyilvántartásban). Bár a rendszer támogatja ezt a leképezést, használata biztonsági okok miatt mégsem ajánlott. Az adminisztrátor bármilyen típusú felhasználói nyilvántartást megjeleníthet az EIM-ben.

Az EIM nem igényli, hogy a meglévő adatokat egy új tárolóba másolja, és megpróbálja a két példányt szinkronban tartani. Az EIM által bevezetett egyetlen új adat a kapcsolat információ. Az adminisztrátorok ezeket az adatokat egy LDAP könyvtárban kezelik, ami lehetővé teszi az adatok egy helyen kezelését, és az információk másolatának használatát, ha erre szükség van.

Kerberos

A Kerberos egy hálózati hitelesítési szolgáltatás, amely lehetőséget nyújt az azonosítók azonosságának ellenőrzésére fizikailag nem biztonságos hálózatokon. A Kerberos kölcsönös hitelesítést, adatbiztonságot és -védelmet nyújt azon feltételezés mellett is, mely szerint a hálózati forgalom elfogható, elemezhető és helyettesíthető.

A Kerberos azonosító egy egyedi azonosító, amely Kerberos hitelesítési szolgáltatásokat használ. A Kerberos megerősíti az azonosságot a hoszt operációs rendszer hitelesítése nélkül, hosztcím alapú megbízhatósággal vagy a hálózaton lévő összes hoszt fizikai biztonságának megkövetelésével.

A Kerberos jegyek olyan meghatalmazások, amelyek az azonosságot ellenőrzik. Kétféle jegy létezik: *jegymegadási jegyek* és *szolgáltatásjegyek*. A jegymegadási jegy a kezdeti azonosítási kérésre születik. Egy hoszt rendszerre való bejelentkezéskor meg kell adni valamit, ami hitelesítheti az azonosságot, például egy jelszót vagy egy tokent. A jegymegadási jegy kézhezvétele után ezzel lehet kérelmezni a különféle szolgáltatásokra vonatkozó szolgáltatásjegyeket. Ezt a kétjegyes módszert nevezzük a Kerberos *megbízható harmadik fél* megközelítésének. A jegymegadási jegy azonosítja a jegy tulajdonosát a Kerberos szerver felé, míg a szolgáltatásjegy a szolgáltatás igénybe vételére feljogosító biztonságos igazolás.

A Kerberos megbízható harmadik fele, vagy köztes személye a *kulcselosztó központ* (KDC). A KDC adja ki az összes Kerberos jegyet a klienseknek.

Biztonságos távoli parancsok áttekintése

A következő információk a biztonságos távoli parancsokkal kapcsolatos részleteket biztosítanak.

Notes:

1. Az Osztott számítási környezet (DCE) 2.2 változatától kezdődően a DCE biztonsági szerver képes Kerberos v5 jegyek visszaadására.
2. Az összes biztonságos távoli parancs (rcmd) a az IBM Hálózati hitelesítési szolgáltatás (NAS) által biztosított, és a bővítőcsomag DVD-n elérhető Kerberos v5 függvénytárat használja. Telepítenie kell a krb5.client.rte fájlkészletet, amely szintén a bővítőcsomag DVD-n érhető el.
3. Ha az AIX operációs rendszert a DVD adathordozó használatával állítja át, és a Kerberos már telepítve van, akkor a telepítési parancsfájl felszólítja a krb5.client.rte telepítésére a bővítőcsomag DVD-ről.
4. Ha az AIX operációs rendszert NIM erőforrások használatával állítja át, és a Kerberos már telepítve van, akkor vegye fel a krb5-öt az lpp_source könyvtárba.

A biztonságos távoli parancsok (rcmd): **rlogin**, **rcp**, **rsh**, **telnet** és**ftp**. A parancsok összefoglaló neve a szabványos AIX módszer. A biztosított kiegészítő módszerek Kerberos módszerek.

A Kerberos v5 hitelesítési módszer használatakor a kliens egy Kerberos v5 jegyet kap a DCE biztonsági szervertől vagy a Kerberos szervertől. A jegy a felhasználó jelenlegi DCE vagy helyi meghatalmazásainak egy része, titkosítva annak a TCP/IP szervernek, amelyhez a felhasználó csatlakozni szeretne. A TCP/IP szerver démonja visszafejti a jegyet. Ez teszi lehetővé a TCP/IP szervernek a felhasználó azonosítását. Ha a jegyen megadott DCE vagy helyi azonosító hozzáférhet az operációs rendszer felhasználói fiókhoz, akkor a kapcsolat folytatódik. A biztonságos távoli parancsok a Kerberos v5 és DCE Kerberos klienseit és szervereit is támogatják.

A kliens hitelesítése mellett a Kerberos 5 továbbítja az aktuális felhasználó meghatalmazásait a TCP/IP szervernek. Ha a meghatalmazások továbbíthatóként vannak megjelölve, akkor a kliens elküldi ezeket a szerverre Kerberos jegymegadási jegyként. Ha a TCP/IP szerver oldalán a felhasználó egy DCE biztonsági szerverrel kommunikál, akkor a démon a **k5dcecreds** paranccsal bővíti a jegymegadási jegyet a teljes DCE meghatalmazások esetén.

Az **ftp** a többi biztonságos távoli parancstól eltérő hitelesítési módszert alkalmaz. Ez a GSSAPI biztonsági mechanizmust használja fel az **ftp** parancs és az **ftpd** démon közötti hitelesítés továbbítására. Az ftp kliens a **clear**, **safe** és **private** részparancsokkal teszi lehetővé az adattitkosítást.

Az operációs rendszer kliensek és szerverek között az **ftp** parancs lehetővé teszi a több byte-os átviteleket a titkosított adatkapcsolatokban. A szabványok a titkosított adatkapcsolatokban csak egybyte-os átviteleket határoznak meg. Ha a felhasználó külső gépekhez csatlakozva használ adattitkosítást, akkor az **ftp** parancs az egybyte-os átviteli korláthoz alkalmazkodik.

Rendszerkonfiguráció:

A biztonságos távoli parancsoknál egy rendszerszintű konfigurációs mechanizmus határozza meg a rendszeren engedélyezett hitelesítési módszereket. A konfiguráció bejövő és kimenő kapcsolatokra is vonatkozik.

A hitelesítési konfiguráció a libauthm.a könyvtárból, valamint az **lsauthent** és **chauthent** parancsokból áll, amelyek lehetővé teszik a **get_auth_methods** és **set_auth_methods** könyvtári rutinok elérését a parancssorból.

A hitelesítési módszer határozza meg, milyen módszerrel hitelesíthetők a hálózati felhasználók. A rendszer a következő hitelesítési módszereket támogatja:

- A Kerberos v5 a legáltalánosabb módszer, mivel ez képezi a DCE alapját is.
- Az rlogin, rsh és rcp biztonságos távoli parancsok támogatják a Kerberos 4. változatát. Ez csak SP rendszereken biztosítja a kompatibilitást a korábbi változatokkal. A Kerberos v4 jegyeket a rendszer nem bővíti fel DCE meghatalmazásokra.

Ha egynél több hitelesítési módszer van beállítva, és az első módszerrel nem sikerül a csatlakozás, akkor a kliens megpróbálkozik a következő beállított hitelesítési módszerrel.

A hitelesítési módszerek tetszőleges sorrendben megadhatók. Az egyetlen megkötés, hogy a szabványos AIX módszernek kell az utolsónak lennie, mivel ez után már nincs visszalépési lehetőség. Ha a szabványos AIX nincs beállítva hitelesítési módszerként, akkor a jelszavas hitelesítés használatára tett kísérleteket a rendszer visszautasítja.

A rendszer hitelesítési módszerek nélkül is beállítható. Ebben az esetben a rendszer az összes biztonságos távoli parancsot használó rendszerkapcsolatot visszautasítja. Emellett, mivel a Kerberos 4. változatát csak az **rlogin**, az **rsh** és az **rcp** parancsok támogatják, a kizárólag Kerberos v4-re beállított rendszerek nem engedélyezik a telnet és ftp kapcsolatokat.

Kerberos v5 felhasználóellenőrzés:

A Kerberos 5. változatú hitelesítési módszerrel érvényesítheti a felhasználókat.

A Kerberos v5 hitelesítési módszer használatakor a TCP/IP kliens egy szolgáltatásjegyet kap, a TCP/IP szerver számára titkosított formában. A szerver a jegy visszafejtésével biztonságos módszerrel tudja azonosítani a felhasználót (DCE vagy helyi azonosító alapján). Ettől függetlenül a szervernek meg kell határoznia, hogy ez a DCE vagy helyi azonosító hozzáférhet-e a helyi fiókhoz. A DCE vagy helyi azonosítónak a helyi operációs rendszer fiókra való leképezését a libvaliduser.a osztott könyvtár végzi, amelyben egyetlen függvény, a kvalid_user található. Ha eltérő leképezési módszerre van szükség, akkor ehhez a rendszeradminisztrátornak kell biztosítania egy alternatív libvaliduser.a könyvtárat.

DCE konfiguráció:

A biztonságos távoli parancsok használatához minden ezekkel elért hálózati csatolónál léteznie kell kettő DCE azonosítónak.

A két DCE azonosító:

```
host/FullInterfaceName
ftp/FullInterfaceName
```

ahol a *FullInterfaceName* a csatoló- és tartománynév

Helyi konfiguráció:

A biztonságos távoli parancsok használatához minden általuk elért hálózati csatolónál léteznie kell két helyi azonosítónak.

A két helyi azonosító:

```
host/csatoló_teljes_neve@tartománynév
ftp/csatoló_teljes_neve@tartománynév
```

ahol a *FullInterfaceName* a csatoló- és tartománynév, a *RealmName* pedig a helyi Kerberos V5 tartomány neve.

Kapcsolódó információkat a következő források tartalmaznak:

- A get_auth_method és a set_auth_method szubrutinok leírása az *Technical Reference: Communications, Volume 2* című kiadványban.
- A *Commands Reference, Volume 1* témakör chauthent paranccsal foglalkozó része
- A *Commands Reference, Volume 3* témakör lsauthent paranccsal foglalkozó része

Hitelesítés az AIX operációs rendszerhez Hálózati hitelesítési szolgáltatással vagy nem AIX szolgáltatásokkal

Az AIX 6.1 változata előtt a KRB5 betöltési modul kezelte a Kerberos hitelesítést a Hálózati hitelesítési szolgáltatás (NAS) környezetben, és a KRB5A betöltési modul kezelte a Kerberos hitelesítést a nem AIX rendszerkörnyezetben. Az AIX 6.1 változatától kezdve a KRB5 betöltési modul kezeli a Kerberos hitelesítést a Hálózati hitelesítési szolgáltatás (NAS) környezetben és a nem AIX rendszerkörnyezetben is. Az etc/security/methods.cfg fájlban található **is_kadmind_compat** attribútum adja meg a KRB5 vagy a KRB5A környezetet. Az AIX 7.1 változatától kezdve a KRB5A betöltési modul nem érhető el. Ezért kötelező az **is_kadmind_compat** attribútum használata az etc/security/methods.cfg fájlban a KRB5 környezet vagy a KRB5A környezet megadására.

Ha a Kerberos kliens be van állítva NAS hitelesítésre, akkor a KRB5 betöltési modul Kerberos hitelesítést és Kerberos azonosítókezelést hajt végre. A modul segítségével a rendszeradminisztrátor kezelheti a Kerberos azonosítókat AIX felhasználóadminisztrációs parancsok alkalmazásával. Azonosítókezelés használatához a Kerberos szervernek támogatnia kell a kadmin adminisztrációs protokollt. A NAS ezt a támogatást a **kadmind** démonnal biztosítja (AIX rendszeren futó Kerberos szerver).

Megjegyzés: A Kerberos kliens beállításakor meg kell adnia, hogy a hitelesítés NAS-on történjen; ellenkező esetben a kliens nem AIX szolgáltatásokon történő hitelesítésre lesz beállítva, és ebben az esetben az azonosítókezelés nem áll rendelkezésre.

Kerberos használatakor nem AIX rendszer esetén a Kerberos azonosítók nem AIX rendszeren vannak tárolva, és nem kezelhetők az AIX operációs rendszerből a kadmin Kerberos adatbázis felületen. Ebben az esetben az azonosítókezelést külön kell végrehajtani a Kerberos azonosítókezelési eszközök segítségével. Ezek az eszközök lehetnek a Kerberos termék részei, vagy integrálhatók operációs rendszerbe (például Windows 2000 rendszerbe). A Kerberos nem AIX rendszerrel valló használatának eredeti célja az volt, hogy hitelesítéstbiztosítson Windows 2000 Active Directory szerverekhez, ahol a Kerberos azonosítókezelés Active Directory kezelőeszközök és API-k használatával valósul meg. A Kerberos nem AIX rendszerekkel azonban használható más szabványnak megfelelő KDC-k esetén is, ahol a Kerberos adminisztrátori felület nem támogatott.

A rendszer telepítése és beállítása Kerberos integrált bejelentkezéshez IBM NAS felhasználásával:

A Hálózati hitelesítési szolgáltatás (NAS) IBM Kerberos megvalósítása a bővítőcsomagban biztosított.

A Kerberos v5 szerver csomagjának telepítéséhez a krb5.server.rte fájlkészletet telepíteni kell a következő parancs futtatásával:

```
installp –aqXYgd . krb5.server
```

Ha a Kerberos szerverként beállítandó gép Kerberos kliensként is alkalmazásra kerül, akkor telepítse a teljes Kerberos KRB5 csomagot.

A DCE a Kerberos segédprogramokkal azonos nevű Kerberos kliens segédprogramok halmazával rendelkezik. A DCE és Kerberos parancsok (pontosabban a **klist**, **kinit** és **kdestroy** parancsok) közötti névtérütközések elkerülése érdekében a Kerberos parancsok az /usr/krb5/bin és /usr/krb5/sbin könyvtárba kerülnek.

A Kerberos parancsok futtatásához meg kell adni a teljes képzésű parancsútvonalat, hacsak nem adja hozzá a Kerberos könyvtárakat a PATH meghatározásához a következőképp:

```
export PATH=$PATH:/usr/krb5/sbin:/usr/krb5/bin
```

Megjegyzés: A Java14 SDK egy **kinit** parancsot is telepít, és ez megelőzhet más **kinit** parancsokat a PATH környezeti változóban. Ha a Hálózati hitelesítési szolgáltatás parancsokra van szükség a Java14 **kinit** program helyett, akkor helyezze át a Java14 **kinit** programot a PATH meghatározás másik helyére.

A Hálózati hitelesítési szolgáltatás dokumentációja a krb5.doc.*nyelv*.pdf|html csomagban található, ahol a *nyelv* a támogatott nyelvre utal.

Az AIX operációs rendszerben két adatbázismoduláll rendelkezésre összetett betöltési modul összeállításához: LDAP és BUILTIN. Az LDAP modul az LDAP nyilvántartásban tárolt adatok (címtár), a BUILTIN modul pedig a fájlok nyilvántartásban tárolt információk (helyi fájlrendszer) eléréséhez használható. A létrehozott összetett betöltési modul neve jellemzően KRB5files vagy KRB5LDAP. Ezek a nevek jelzik, hogy a KRB5 hitelesítéshez és a helyi fájlokhoz, vagy LDAP-hez kerül felhasználásra.

A Hálózati hitelesítési szolgáltatás a Kerberos információk helyi fájlrendszeren (Kerberos örökölt adatbázis) és LDAP-ben történő tárolását is támogatja. Négy lehetséges konfiguráció létezik:

- KRB5files Kerberos örökölt adatbázisban tárolt Kerberos szerverinformációkkal
- KRB5files Kerberos LDAP adatbázisban tárolt Kerberos szerverinformációkkal
- KRB5LDAP KRB5files Kerberos örökölt adatbázisban tárolt Kerberos szerverinformációkkal
- KRB5LDAP Kerberos LDAP adatbázisban tárolt Kerberos szerverinformációkkal

Ha LDAP a tárolómechanizmus Kerberos azonosítók vagy AIX felhasználói- és csoportinformációk tárolásához, akkor végezze el az LDAP beállítást még mielőtt meghívja a Kerberos konfigurációs parancsokat. Az LDAP beállítása után az **mkkrb5srv** parancs segítségével állítsa be a Kerberos szervereket.

Hálózati hitelesítési szolgáltatás beállítása örökölt adatbázis-tárolóval:

A Hálózati hitelesítési szolgáltatás KDC és az adminisztrációs szerverek beállíthatók örökölt Kerberos adatbázissal és beállíthatja a Hálózati hitelesítési szolgáltatás szervereket az **mkkrb5srv** parancs segítségével.

Az **mkkrb5srv** paranccsal kapcsolatos további információkért tekintse meg az **mkkrb5srv** parancsot.

Megjegyzés: A DCE és Kerberos szerverszoftvert ne telepítse ugyanarra a fizikai rendszerre. Ha erre mégis mindenképpen szükség van, akkor a DCE kliens és szerver, vagy a Kerberos kliens és szerver portszámait módosítani kell. Bármelyik lehetőséget választja is, egy ilyen változás jelentősen befolyásolhatja a környezet meglévő DCE és Kerberos szolgáltatásaival való együttműködést. A DCE és Kerberos együttéléséről a Hálózati hitelesítési szolgáltatás dokumentációban talál további információkat.

A Kerberos 5. változata úgy van beállítva, hogy visszautasítsa minden olyan hoszt jegyét, amelynek a kulcselosztó központhoz (KDC) képest vett időeltérése nagyobb a megadott maximális értéknél. Az időeltérés alapértelmezett értéke 300 másodperc (5 perc). A Kerberos valamilyen formában megköveteli a szerverek és kliensek órájának összehangolását. Az idő összehangolására az **xntpd** vagy **timed** démonok használata javasolt. A **timed** démon használatához tegye a következőket:

1. A KDC szervert állítsa be időszervernek a **timed** démon indításával az alábbiak szerint:
   ```
   timed -M
   ```
2. Indítsa el a **timed** démont minden egyes Kerberos kliensen a következőképp:
   ```
   timed -t
   ```
3. A Kerberos KDC és kadmin szerverek beállításához futtassa az **mkkrb5srv** parancsot. Ha például a Kerberos szolgáltatásokat a MYREALM tartományra, a sundial szerverre és az xyz.com DNS tartományra kívánja beállítani, akkor futtassa a következő parancsot:
   ```
   mkkrb5srv -r MYREALM -s sundial.xyz.com -d xyz.com -a admin/admin
   ```

Várjon néhány percet, amíg az /etc/inittab könyvtárból a **kadmind** és **krb5kdc** parancs elindításra kerül.

Hálózati hitelesítési szolgáltatás a **/var** fájlrendszer területét használja az információk tárolásához. Ezek az információk magukban foglalják a hitelesített felhasználó adatbázis-, napló- és hitelesítési adat ideiglenes tárolófájljait. A fájlok mérete az idő múlásával növekedhet. Győződjön meg róla, hogy a **/var** fájlrendszer elegendő szabad területtel rendelkezik ezen információk tárolásához a szabad terület rendszeres időközönkénti figyelésével.

A következő egy tipikus **mkkrb5srv** parancs:

```
mkkrb5srv –r Tartomány_Neve –s KDC_szerver –d Tartomány_Neve –a Admin_Neve
```

A 16. táblázat: változó értékei kerülnek felhasználásra a következő példában annak bemutatására, hogy a Hálózati hitelesítési szolgáltatás szerverek hogyan állíthatók be örökölt adatbázissal.

16. táblázat: **mkkrb5srv** *parancs változónevei*

Változó neve	Változó értéke
Realm Name	MYREALM
KDC Server	kdcsrv.austin.ibm.com
Domain Name	austin.ibm.com
Administrator Name	admin/admin

Ha van meglévő Kerberos szerverkonfiguráció, akkor azt eltávolíthatja az **mkkrb5srv –U** vagy **unconfig.krb5** parancs segítségével.

FIGYELEM: Ha meg kell tartania egy meglévő Kerberos szerverkonfigurációt, akkor ne hajtsa végre a következő lépéseket.

A következő eljárás bemutatja, hogy a Hálózati hitelesítési szolgáltatás szerverek hogyan állíthatók be örökölt adatbázissal.

1. Adja ki a következő parancsot:

   ```
   mkkrb5srv -r MYREALM -s kdcsrv.austin.ibm.com -d austin.ibm.com -a admin/admin
   ```

   A parancs kiadása után meg kell adnia az elsődleges adatbázisjelszót.

   Mivel a Hálózati hitelesítési szolgáltatás nem támogatja azokat a konfigurációkat, amelyben a KDC és az adminisztrációs szerver másik hoszton található, a helyi hoszt kerül felhasználásra a KDC és az adminisztrációs szerverhez egyaránt. Hagyja figyelmen kívül a következő hibaüzenetet, ha megjelenik: The -s option is not supported.
2. Adja meg az elsődleges adatbázisjelszót, amikor a rendszer felszólítja erre.
3. Adja meg az adminisztrációs azonosító jelszót, amikor a rendszer felszólítja erre.

   Az adminisztrációs azonosító jelszó megadása után az **mkkrb5srv** parancs elindítja a **kadmind** és **krb5kdc** démont az /etc/inittab fájlútvonalról. Ez a folyamat eltarthat pár percig.
4. Ellenőrizze az /etc/inittab fájl bejegyzéseit a következő parancsok futtatásával:

   ```
   lsitab krb5kdc
   lsitab kadm
   ```
5. Ellenőrizze, hogy a KDC és a kadmind szerver elindult-e a következő parancs beírásával:

   ```
   ps -ef | grep -v grep | grep krb5
   ```

Az **mkkrb5srv** parancs létrehozza az elsődleges KDC és a kadmind adminisztrációs szervert a Kerberos tartományhoz (MYREALM). A konfigurációs fájlokat is létrehozza, inicializálja az azonosító-adatbázist és elindítja a KDC és kadmind szervert.

Az **mkkrb5srv** parancs a következő tevékenységeket hajtja végre:

1. Létrehozza az /etc/krb5/krb5.conf fájlt. A tartomány nevét, a Kerberos adminisztrációs szervert és a DNS tartománynevet a parancssorból veszi át. A /etc/krb5/krb5.conf fájl emellett a default_keytab_name, kdc és admin_server naplófájlok elérési útjait is beállítja.
2. Létrehozza a /var/krb5/krb5kdc/kdc.conf fájlt. A /var/krb5/krb5kdc/kdc.conf fájl beállítja a *kdc_ports*, *kadmin_port*, *max_life*, *max_renewable_life*, *master_key_type* és *supported_enctypes* változók értékeit. A fájl emellett megadja a *database_name*, *admin_keytab*, *acl_file*, *dict_file* és *key_stash_file* változók elérési útjait is.
3. Létrehozza a /var/krb5/krb5kdc/kadm5.acl fájlt. Ez adja meg az admin, root és host azonosítók hozzáférését.
4. Létrehozza az adatbázist és egy admin azonosítót. A rendszer megkéri egy Kerberos mesterkulcs beállítására, valamint a Kerberos adminisztrátori azonosság megnevezésére és jelszavának megadására. Katasztrófa utáni helyreállítási szempontból rendkívül fontos, hogy a mesterkulcs és az adminisztrátori azonosság illetve jelszó biztonságos helyen legyen.

További információkat a “Példa futások” oldalszám: 289 és “Hibaüzenetek és helyreállítási tevékenységek” oldalszám: 288 tartalmaz.

Kerberos szerver beállítása LDAP tárolóval:

Beállíthatja a Hálózati hitelesítési szolgáltatás kadmint és a KDC szervereket Kerberos integrált bejelentkezésre az **mkkrb5srv** parancs segítségével.

A 17. táblázat: változóértékei kerülnek felhasználásra a következő példában annak bemutatására, hogy a Hálózati hitelesítési szolgáltatás szerverösszetevők hogyan állíthatók be LDAP tárolóval az **mkkrb5srv** parancs segítségével.

17. táblázat: **mkrb5srv** *parancs változónevei*

Változó neve	Változó értéke
Realm_Name	MYREALM
KDC_Server	kdcsrv.austin.ibm.com
Domain_Name	austin.ibm.com
Admin_Name	admin/admin
LDAP szerver	kdcsrv.austin.ibm.com
LDAP administrator name	cn=root
LDAP administrator password	secret

A következő eljárás bemutatja, hogy a Hálózati hitelesítési szolgáltatás szerverösszetevők hogyan állíthatók be LDAP tárolóval az **mkkrb5srv** parancs segítségével.

1. Futtassa a következő parancsot:
```
mkkrb5srv -r MYREALM -s kdcsrv.austin.ibm.com -d austin.ibm.com\
-a admin/admin -l kdcsrv.austin.ibm.com -u cn=root -p secret
```
2. A következő parancs futtatásával ellenőrizze, hogy a KDC és a kadmind szerver elindult-e:
```
ps -ef | grep -v grep | grep krb5
```

Az **mkkrb5srv** parancs futtatása LDAP-vel a parancs örökölt adatbázis-konfigurációval történő futtatásához hasonló eredményeket állít elő. LDAP használata esetén azonban az adatbázisok nem a helyi fájlrendszeren jönnek létre. Ehelyett egy .kdc_ldap_data fájl kerül létrehozásra a /var/krb5/krb5kdc fájlban az LDAP-vel kapcsolatos információk tárolása érdekében.

A használattal kapcsolatos információkért tekintse meg az **mkkrb5srv** parancsot.

Kerberos integrált bejelentkezés beállítása:

A Kerberos telepítés befejezése után be kell állítani a rendszert, hogy a Kerberost használja elsődleges felhasználói hitelesítésként.

Ha egy rendszeren A Kerberost elsődleges felhasználó hitelesítési módszernek kívánja beállítani, akkor futtassa az **mkkrb5clnt** parancsot a következő paraméterekkel:

```
mkkrb5clnt -c KDC -r tartomány -a adminisztrátor -s szerver -d tartomány -A -i adatbázis -K -T
```

Az 18. táblázat: változóértékei kerülnek felhasználásra az alábbi példában annak bemutatása érdekében, hogy hogyan kell konfigurálni egy rendszert Kerberos integrált bejelentkezésre helyi fájlrendszerhez AIX felhasználó/csoport lerakatként.

18. táblázat: **mkkrb5clnt** *parancs változónevei*

Változó neve	Változó értéke
Terület neve	MYREALM
KDC szerver	kdcsrv.austin.ibm.com
Tartománynév	austin.ibm.com
Felügyeleti szerver	kdcsrv.austin.ibm.com
Adminisztrátor neve	admin/admin
AIX felhasználó/csoport adatbázis	files

A következő parancs példája bemutatja egy rendszer beállítását Kerberos integrált bejelentkezésre helyi fájlrendszerhez AIX felhasználó/csoport lerakatként.

Futtassa a következő parancsot:

```
mkkrb5clnt -r MYREALM -c kdcsrv.austin.ibm.com -s kdcsrv.austin.ibm.com\
-a admin/admin -d austin.ibm.com -A -i files -K -T
```

Az előző példa a következő tevékenységeket végzi el:

1. A parancs létrehozza az /etc/krb5/krb5.conf fájlt. A tartomány nevét, a Kerberos adminisztrációs szervert és a DNS tartománynevet a parancssorból veszi át. A default_keytab_name, kdc és kadmin naplófájlok elérési útja szintén frissítésre kerül.
2. A **-i** kapcsoló állítja be a teljesen integrált bejelentkezést. A megadott adatbázis tartalmazza az AIX felhasználói azonosításra vonatkozó információkat. Ez nem egyezik meg a Kerberos azonosítók tárolási tárolójával. A Kerberos azonosítók tárolásának helyét a Kerberos konfiguráció során lehet beállítani.
3. A **-K** kapcsoló állítja be a Kerberost alapértelmezett hitelesítési sémaként. Ez teszi lehetővé, hogy a felhasználók hitelesítését a bejelentkezéskor a Kerberos lássa el.
4. A **-A** kapcsoló felvesz egy bejegyzést a Kerberos adatbázisban, hogy a root felhasználó Kerberos adminisztrátor legyen.
5. A **-T** kapcsoló kéri le a szerver adminisztrátor jegymegadási jegyet.

Megjegyzés: Ne használja a **-D** paramétert az **mkkrb5clnt** paranccsal a Kerberos kliensкörnyezet beállításához IBM Hálózati hitelesítési szolgáltatás (NAS) esetén. Ha nem adja meg a **-D** paramétert az **mkkrb5clnt** parancsban, akkor az **is_kadmind_compat** attribútum nem szerepel a /usr/lib/security/methods.cfg fájlban, és a Kerberos kliensкörnyezet IBM NAS hitelesítéshez lesz beállítva.

Ellenőrizze a konfigurációt az /etc/krb5/krb5.conf fájl megtekintésével. A következő a kliensgépen lévő /etc/krb5/krb5.conf fájlra példa:

```
[libdefaults]
     default_realm = MYREALM
     default_keytab_name = FILE:/etc/krb5/krb5.keytab
     default_tkt_enctypes = des3-cbc-sha1 arcfour-hmac aes256-cts des-cbc-md5 des-cbc-crc
     default_tgs_enctypes = des3-cbc-sha1 arcfour-hmac aes256-cts des-cbc-md5 des-cbc-crc
[realms]
     MYREALM = {
        kdc = kdcsrv.austin.ibm.com:88
        admin_server = kdcsrv.austin.ibm.com:749
        default_domain = austin.ibm.com
```

```
    }
[domain_realm]
    .austin.ibm.com = MYREALM
    kdcsrv.austin.ibm.com = MYREALM
[logging]
     kdc = FILE:/var/krb5/log/krb5kdc.log
     admin_server = FILE:/var/krb5/log/kadmin.log
     default = FILE:/var/krb5/log/krb5lib.log
```

Megjegyzés: Ha az LDAP kerül felhasználásra a Kerberos azonosítótárolóhoz, akkor a krb5.conf fájl tartalmazni fogja a következő sort a [realms] szakasz alatt:

```
vdb_plugin_lib = /usr/lib/libkrb5ldplug.a
```

Ha a telepített rendszer a kulcselosztó központtól eltérő DNS tartományban található, akkor a következő további lépések elvégzése szükséges:

1. Az /etc/krb5/krb5.conf fájlban adjon hozzá egy másik bejegyzést a [domain realm] után.
2. Képezze le a másik DNS tartományt a Kerberos tartományra.

Ha például a MYREALM tartományban el kíván helyezni egy olyan klienst is, amelynek DNS tartománya abc.xyz.com, akkor módosítsa az /etc/krb5/krb5.conf fájlt a következőképp:

```
[domain realm]
     .austin.ibm.com = MYREALM
     .raleigh.ibm.com = MYREALM
```

Ha a Hálózati hitelesítési szolgáltatás konfigurációja befejeződött, akkor az operációs rendszer bejelentkezési folyamata változatlan marad. Sikeres bejelentkezés után a felhasználók futó folyamataihoz Kerberos jegymegadási jegy kerül hozzárendelésre. A felhasználó $KRB5CCNAME környezeti változója erre a jegymegadási jegyre mutat. A **klist** parancs segítségével ellenőrizze, hogy a bejelentkezés sikeres volt-e és a felhasználó rendelkezik-e jegymegadási jeggyel.

Megjegyzés: Az **mkkrb5clnt** parancs futtatásakor a következő szakasz kerül a methods.cfg fájlba.

```
KRB5:
        program = /usr/lib/security/KRB5
        program_64 = /usr/lib/security/KRB5_64
        options = is_kadmind_compat=yes

KRB5files:
        options = db=BUILTIN,auth=KRB5
```

További információkat

- az **mkkrb5clnt** parancsról itt talál: **mkkrb5clnt** parancs.
- a methods.cfg fájlról itt talál: methods.cfg fájl.

Hibaüzenetek és helyreállítási tevékenységek:

Az **mkkrb5srv** parancs kapcsán felmerülő lehetséges hibák a következők:

- Ha a krb5.conf, kdc.conf, vagy kadm5.acl fájl már létezik, akkor az **mkkrb5srv** parancs nem módosítja az értékeket. Ehelyett egy üzenet fogja jelezni, hogy a fájlok már léteznek. A konfigurációs értékek a krb5.conf, kdc.conf vagy kadm5.acl fájlok szerkesztésével módosíthatók.
- Ha elgépelt valamit, és nem jött létre adatbázis, akkor távolítsa el a létrehozott konfigurációs fájlokat, majd futtassa ismét a parancsot.
- Ha következetlenség áll fenn az adatbázis és a konfigurációs értékek között, akkor távolítsa el az adatbázist a /var/krb5/krb5kdc/* könyvtárból, majd futtassa ismét a parancsot.
- Győződjön meg róla, hogy a **kadmind** és a **krb5kdc** démonok elindultak a számítógépen. A démonok futásának ellenőrzésére használja a **ps** parancsot. Ha a démonok nem indultak el, akkor nézze meg a naplófájlt.

Az **mkkrb5clnt** parancs kapcsán felmerülő lehetséges hibák a következők:

- A krb5.conf helytelen értékei az /etc/krb5/krb5.conf fájl módosításával javíthatók.
- A **-i** kapcsoló helytelen értékei a /usr/lib/security/methods.cfg fájl módosításával javíthatók ki.

Kadmind démon függőség megszüntetése nem KRB5 hitelesítés során: A KRB5 betöltési modul késleltetést okoz, amikor a kadmind démon nem érhető el és amikor nem KRB5 hitelesítési mechanizmust, például egypontos bejelentkezést (SSO) használ. A függőség megszüntetéséhez állítsa be a kadmind_timeout paramétert a **methods.cfg** fájlban.

A lehetséges értékek: kadmind_timeout=<másodpercek száma>, ahol a másodpercek számának 0-nál nagyobbnak kell lennie.

Amikor a KRB5 betöltési modul egy nem működő kadmind kiszolgálóhoz próbál csatlakozni, akkor Átvitelvezérlési protokoll (TCP) időtúllépés történik. A kadmind_timeout paraméter megakadályozza a további késleltetést a kezdeti TCP időtúllépés után. A kadmind_timeout paraméter megadja, hogy a KRB5 betöltési modul milyen időablakban próbálkozzon új kadmind csatlakozással a kezdeti TCP időtúllépés után. Amikor a kadmind kiszolgáló fut, akkor továbbra is az alapértelmezett viselkedés van érvényben.

Alapértelmezésben a kadmind_timeout tiltott. A kadmind_timeout paraméter engedélyezéséhez módosítsa a methods.cfg fájlt az alábbiak szerint:

```
KRB5:
        program = /usr/lib/security/KRB5
        options = kadmind_timeout=300
KRB5files:
        options = db=BUILTIN,auth=KRB5
```

Létrehozott fájlok:

Az **mkkrb5srv** parancs a következő fájlokat hozza létre:

- /etc/krb5/krb5.conf
- /var/krb5/krb5kdc/kadm5.acl
- /var/krb5/krb5kdc/kdc.conf

Az **mkkrb5clnt** parancs a következő fájlokat hozza létre:

- /etc/krb5/krb5.conf

Az **mkkrb5clnt -i** *files* kapcsoló a következő szakaszokat adja hozzá az /usr/lib/security/methods.cfg fájlhoz:

```
KRB5:
  program =
  options =
KRB5files:
  options =
```

Példa futások:

Ez a rész példa futásokból tartalmaz példákat.

Az alábbi lista az **mkkrb5srv** parancs kimenetére mutat be egy példát:

```
# mkkrb5srv -r MYREALM -s sundial.xyz.com -d xyz.com -a admin/admin
```

Az alábbihoz hasonló kimenet jelenik meg:

```
  Fileset                      Level  State      Description
  ----------------------------------------------------------------------------
Path: /usr/lib/objrepos
  krb5.server.rte            1.3.0.0  COMMITTED  Network Authentication Service
                                                 Server
```

```
Path: /etc/objrepos
  krb5.server.rte            1.3.0.0  COMMITTED  Network Authentication Service
                                                 Server

The -s option is not supported.
The administration server will be the local host.
Initializing configuration...
Creating /etc/krb5/krb5.conf...
Creating /var/krb5/krb5kdc/kdc.conf...
Creating database files...
Initializing database '/var/krb5/krb5kdc/principal' for realm 'MYREALM'
master key name 'K/M@MYREALM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter database Master Password:
Re-enter database Master Password to verify:
WARNING: no policy specified for admin/admin@MYREALM;
  defaulting to no policy. Note that policy may be overridden by
  ACL restrictions.
Enter password for principal "admin/admin@MYREALM":
Re-enter password for principal "admin/admin@MYREALM":
Principal "admin/admin@MYREALM" created.
Creating keytable...
Creating /var/krb5/krb5kdc/kadm5.acl...
Starting krb5kdc...
krb5kdc was started successfully.
Starting kadmind...
kadmind was started successfully.
The command completed successfully.
Restarting kadmind and krb5kdc
```

Az alábbi lista az **mkkrb5clnt** parancs kimenetére mutat be egy példát:

```
mkkrb5clnt -r MYREALM -c sundial.xyz.com -s sundial.xyz.com \
           -a admin/admin -d xyz.com -i files -K -T -A
```

Az alábbihoz hasonló kimenet jelenik meg:

```
Initializing configuration...
Creating /etc/krb5/krb5.conf...
The command completed successfully.
Password for admin/admin@MYREALM:
Configuring fully integrated login
Authenticating as principal admin/admin with existing credentials.
WARNING: no policy specified for host/diana.xyz.com@MYREALM;
  defaulting to no policy. Note that policy may be overridden by
  ACL restrictions.
Principal "host/diana.xyz.com@MYREALM" created.

Administration credentials NOT DESTROYED.
Authenticating as principal admin/admin with existing credentials.

Administration credentials NOT DESTROYED.
Authenticating as principal admin/admin with existing credentials.
Principal "kadmin/admin@MYREALM" modified.

Administration credentials NOT DESTROYED.
Configuring Kerberos as the default authentication scheme
Making root a Kerberos administrator
Authenticating as principal admin/admin with existing credentials.
WARNING: no policy specified for root/diana.xyz.com@MYREALM;
  defaulting to no policy. Note that policy may be overridden by
  ACL restrictions.
Enter password for principal "root/diana.xyz.com@MYREALM":
Re-enter password for principal "root/diana.xyz.com@MYREALM":
```

```
Principal "root/diana.xyz.com@MYREALM" created.

Administration credentials NOT DESTROYED.
Cleaning administrator credentials and exiting.
```

A hitelesítés kadmind démontól függésének megszüntetése:

A KRB5 betöltési modul hitelesítése sikertelen akkor, ha a **kadmind** démon nem elérhető. A függőség a methods.cfg fájl *kadmind* paraméterének beállításával megszüntethető.

Lehetséges értékek: kadmind=no és kadmind=false a **kadmind** kikeresések letiltásához, illetve kadmind=yes és kadmind=true a **kadmind** kikeresések engedélyezéséhez (az alapértelemzett érték a yes). Ha a beállítás értéke no, akkor a **kadmind** démon a hitelesítés során nem kerül csatlakoztatásra. Így a felhasználók a **kadmind** démon állapotától függetlenül bejelentkezhetnek feltéve, hogy helyes jelszót adnak meg a rendszer kérdésére. Azonban az AIX felhasználóadminisztrációs parancsok, mint az **mkuser**, a **chuser** vagy az **rmuser** nem fog működni Kerberos integrált felhasználók adminisztrációjához, ha a démon nem érhető el (például a démon nem fut, vagy a számítógép nem érhető el).

A *kadmind* paraméter alapértelmezett értéke yes. Ez azt jelenti, hogy a hitlesítés megkísérli a **kadmind** kikereséseket. Alapértelmezett esetben ha a démon nem elérhető, akkor a hitelesítés tovább tarthat.

A **kadmind** démon ellenőrzésének letiltásához a hitelesítés során módosítsa a methods.cfg megfelelő szakaszait az alábbiak szerint:

```
KRB5:
        program = /usr/lib/security/KRB5
        options = kadmind=no
KRB5files:
        options = db=BUILTIN,auth=KRB5
```

Ha a **kadmind** démon nem elérhető, akkor a root felhasználó nem képes megváltoztatni a felhasználók jelszavait. Ha egy felhasználó elfelejti a jelszavát, akkor elérhetővé kell tenni a **kadmind** démont. Továbbá ha a felhasználó Kerberos azonosítónevet ad meg a bejelentkezés során, akkor az azonosító elsődleges neve meg lesz rövidítve az AIX felhasználónév hosszkorlátozásának megfelelően. Ez a megrövidített név kerül felhasználásraaz AIX felhasználóazonosítási információk lekéréséhez (például a saját könyvtár érték lekéréséhez).

Ha a **kadmind** démon nem elérhető (a démon nem fut vagy nem létesíthető vele kapcsolat), akkor a **mkuser** parancs az alábbi hibát adja:

```
3004-694 Error adding "krb5user": You do not have permission.
```

Ha a *kadmind* paraméter értéke no vagy a **kadmind** démon nem elérhető, akkor a rendszer nem tudja megerősíteni az azonosító meglétét a Kerberos adatbázisban, így nem kéri le a Kerberosszal kapcsolatos attribútumokat. Ez a helyzet hiányos vagy pontatlan eredményt ad. Elképzelhető, hogy az **lsuser** parancs például nem jelent felhasználókat az ÖSSZES lekérdezéshez.

Ezen felül a **chuser** parancs csak az AIX-hez tartozó attribútumokat kezeli, a Kerberoshoz tartozókat nem. Az **rmuser** nem képes Kerberos azonosítók törlésére, és a **passwd** parancs sikertelen a Kerberos által hitelesített felhasználók esetében.

Ha nem elérhető az a hálózat, amelyben a **kadmind** démon található, akkor a válaszidő jelentős késlekedést okozhat. Ha a gép nem elérhető, akkor a hitelesítés késleltetésének megszüntetése érdekében állítsa a methods.cfg fájl kadmind paraméterét no értékre.

Ha a **kadmind** nem fut, akkor a lejárt jelszavú felhasználók nem jelentkezhetnek be és nem cserélhetik le a jelszavukat.

Ha a kadmind=no érték be van állítva, de a **kadmind** démon fut, akkor futtathatja a következő parancsokat: **login**, **su**, **passwd**, **mkuser**, **chuser** és **rmuser**.

Kerberos a Hálózati hitelesítési szolgáltatáson: hibaelhárítási információk:

Ez hibaelhárítási információkat biztosít a Kerberos kliensekről, melyek Kerberos szervert használnak AIX operációs rendszeren.

Az LDAP modul hiba- és hibakeresési információkat ír a syslog alrendszerbe.

Az IBM Hálózati hitelesítési szolgáltatás a saját naplófájljait használja a KDC és **kadmind** démonhoz intézett kérések naplózásához. A naplófájlok a krb5.conf fájl [logging] szakaszában vannak megadva. A fájlok alapértelmezett helye: /var/krb5/log/krb5kdc.log és /var/krb5/log/kadmin.log .

Ha a probléma a IBM Tivoli Directory Serverrel kapcsolatos, akkor ellenőrizze az IBM Tivoli Directory Server által előállított naplófájlokat. Alapértelmezésben a naplófájlok a következő helyen találhatók: /var/ldap/ibmslapd.log és /var/ldap/db2cli.log.

- ***Hogyan hozhatok létre AIX Kerberos hitelesítésű felhasználókat?***

  A root felhasználónak be kell szereznie a Kerberos hitelesítési adatokat, amelyek biztosítják a szükséges jogosultságot az adminisztrációs feladatok végrehajtásához. Az adminisztrációs feladatok a következő KDC szerveren kerülnek végrehajtásra: kdcsrv.austin.ibm.com.

  Hozzon létre egy AIX felhasználói fiókot (foo) és Kerberos azonosítót (foo@MYREALM) a Kerberos adatbázison a következő parancsokkal:

  ```
  kinit root/kdcsrv.austin.ibm.com
  mkuser –R KRB5files SYSTEM=KRB5files registry=KRB5files foo
  ```

  Ezek a parancsok szintén hitelesítik a felhasználót a KRB5files fájlokhoz.

  Ha az LDAP beállítást az **mksecldap** paranccsal végezte el, akkor a következő parancsokkal hozhat létreAIX Kerberos hitelesítésű felhasználókat:

  ```
  mkuser –R KRB5LDAP SYSTEM=KRB5LDAP registry=KRB5LDAP foo
  ```

- ***Hogyan távolítható el a Kerberos hitelesített felhasználó?***

  Kerberos hitelesített felhasználó eltávolításához adja ki a következő parancsot:

  ```
  rmuser –R KRB5files foo
  ```

  Ha az LDAP-t az **mksecldap** paranccsal állította be, akkor eltávolíthatja a Kerberos hitelesített felhasználót a következő parancs kiadásával:

  ```
  rmuser –R KRB5LDAP foo
  ```

- ***Hogyan módosítható egy Kerberos hitelesített felhasználó jelszava?***

  Kerberos hitelesített felhasználó jelszavának módosításához adja ki a következő parancsot:

  ```
  passwd –R KRB5files foo
  ```

- ***Mik azok az AIX Kerberos kiterjesztett attribútumok?***

  A Kerberos azonosítóinformációk AIX kiterjesztett attribútumokkal kezelhetők az AIX **lsuser** és **chuser** parancsai használatával. Csak a GET hozzáférési móddal rendelkező attribútumok jeleníthetők meg. Azon attribútumokhoz, melyek esetében a SET hozzáférési mód van beállítva, privilegizált felhasználó rendelhet értéket (root felhasználó az AIX operációs rendszeren). AzAIX Kerberos hitelesítésű felhasználó megjelenítheti a saját Kerberos kiterjesztett attribútumait és más engedélyezett AIX attribútumokat, úgymint: pgrp, groups, gecos, home és shell.

  A 19. táblázat: oldalszám: 293 felsorolja azAIX Kerberos kiterjesztett attribútumokat és azok hozzáférési módjait.

19. táblázat: AIX Kerberos kiterjesztett attribútumok és hozzáférési módok

Kiterjesztett attribútum neve	Leírás	Hozzáférési mód
krb5_principal_name	Az AIX felhasználónévhez tartozó azonosítónév.	GET
krb5_principal	Megegyezik a krb5_principal_name attribútummal.	GET
krb5_realm	A Kerberos tartomány neve, amelyhez az azonosító tartozik.	GET
krb5_last_pwd_change	Az azonosítóhoz tartozó jelszó utolsó módosítási ideje.	GET
krb5_attributes	A KDC által használt attribútumok halmaza.	GET/SET
krb5_mod_name	Az azonosítót utoljára módosító felhasználó neve.	GET
krb5_mod_date	Az azonosító utolsó módosításának ideje.	GET
krb5_kvno	Az azonosító aktuális kulcsának (jelszó) verziószáma.	GET/SET
krb5_mkvno	Az adatbázis-mesterkulcs verziószáma. Ez más megvalósításokkal való kompatibilitás érdekében biztosított. A mező értéke 0.	GET
krb5_max_renewable_life	Az azonosítóhoz kiadott jegy maximális megújítható élettartama.	GET/SET
krb5_names	A név:hosztnév párok listája. Ez a mező jövőbeli használatra készült. Ne módosítsa az attribútumot.	GET/SET

A krb5_attributes kiterjesztett attribútum a KDC által használható Kerberos azonosítóattribútumok halmazát ábrázolja. A jogosult felhasználó a **chuser** parancs segítségével módosíthatja a ezeket a Kerberos attribútumokat.

```
chuser –R KRB5files krb5_attributes=+requires_preauth krb5user
```

Paraméter beállításához írjon plusz (+) jelet a paraméter elé. Paraméter visszaállításához írjon mínusz (-) jelet a paraméter elé. Például:

+*attribute_name* beállítja a paramétert

-*attribute_name* visszaállítja a paramétert

Megjegyzés: Felhasználó létrehozásakor a következők kivételével az összes attribútum beállításra kerül: requires_hwauth, needchange, password_changing_service és support_desmd5

A következő lista a krb5_attributes kiterjesztett attribútum attribútumait tartalmazza:

allow_postdated
Ha be van állítva, akkor az azonosítóhoz kiadhatók utódátumozott jegyek.

allow_forwardable
Ha be van állítva, akkor az azonosítóhoz kiadhatók továbbítható jegyek.

allow_tgs_req
Ha be van állítva, akkor az azonosítóhoz tartozó szervizjegyek jegymegadási jeggyel kerülnek kiadásra.

allow_renewable
Ha be van állítva, akkor az azonosítóhoz kiadhatók megújítható jegyek.

allow_proxiable
Ha be van állítva, akkor az azonosítóhoz kiadhatók átadható jegyek.

allow_dup_skey
Ha be van állítva, akkor a felhasználók közötti hitelesítés engedélyezett az azonosítóhoz.

allow_tix
Ha be van állítva, akkor jegyek kerülnek kiadásra az azonosítóhoz.

requires_preauth
Ha be van állítva, akkor a szoftver előzetes hitelesítése szükséges a jegy kiadása előtt.

requires_hwauth
Ha be van állítva, akkor a hardver szoftver általi előzetes hitelesítése szükséges, mielőtt jegy kerülne kiadásra az azonosítóhoz.

needchange
Ha be van állítva, akkor az azonosítóhoz tartozó kulcsot (jelszó) jegyek kiadása előtt módosítani kell.

Megjegyzés: Ha a needchange jelző be van állítva, akkor a felhasználót a rendszer felszólítja a jelszó módosítására a következő bejelentkezési kísérlet során. Ebben az esetben a felhasználó hitelesítve lesz (Kerberosszal), de nem fog rendelkezni jegymegadási jeggyel. Jegymegadási jegy eléréséhez a felhasználónak meg kell hívnia a **kinit** parancsot. A needchange jelző csak Hálózati hitelesítési szolgáltatás modult használó Kerberosra érvényes.

allow_svr
Ha be van állítva, akkor a szervizjegyek kiadhatók az azonosítóhoz.

password_changing_service
Ha be van állítva, akkor az azonosító a jelszómódosítási szolgáltatás speciális azonosítója

support_desmd5
Ha be van állítva, akkor a KDC kiadhat RSA MD5 ellenőrző összeg algoritmust használó jegyeket.

Megjegyzés: Az attribútum beállítása együttműködési problémákhoz vezethet.

- ***Hogyan listázhatom ki az AIX Kerberos kiterjesztett attribútumokat?***

  Az AIX Kerberos kiterjesztett attribútumok megjelenítése érdekében futtassa a következő parancsot:

  ```
  lsuser –R KRB5files foo
  ```

  Az –a paraméterrel megjelenítheti a specifikus kiterjesztett attribútumokat. Például:

  ```
  lsuser -R KRB5files -f -a krb5_principal krb5_principal_name krb5_realm
  ```

- ***Hogyan módosíthatom az AIX Kerberos kiterjesztett attribútumokat?***

  Csak a jogosult felhasználó módosíthatja a következő SET hozzáférési módú kiterjesztett attribútumokat: krb5_kvno, krb5_max_renewable_life, krb5_attributes és krb5_names.

  – A foo számára kiadott jegyek maximális megújítási élettartamának öt napra módosításához adja ki a következő parancsot:

    ```
    chuser -R KRB5files krb5_max_renewable_life=432000 foo
    ```

  – A foo felhasználóhoz tartozó azonosító kulcs (jelszó) verziószámának módosításához adja ki a következő parancsot:

    ```
    chuser -R KRB5files krb5_kvno=4 foo
    ```

  – A 19. táblázat: oldalszám: 293 által felsorolt összes Kerberos azonosító attribútum beállításához adja ki a következő parancsot:

    ```
    chuser -R KRB5files krb5_attributes=+allow_postdated,+allow_forwardable,\
    +allow_tgs_req,+allow_renewble,+allow_proxiable,+allow_dup_skey,+allow_tix,\
    +requires_preauth,+requires_hwath,+needchange,+allow_svr,\
    +password_changing_service,+support_desmd5 foo
    ```

    ```
    lsuser -R KRB5files -a krb5_attributes foo
    ```

  – A 19. táblázat: oldalszám: 293 által felsorolt összes Kerberos azonosító attribútum visszaállításához adja ki a következő parancsot:

    ```
    chuser -R KRB5files krb5_attributes=-allow_postdated,-allow_forwardable,\
    -allow_tgs_req,-allow_renewable,-allow_proxiable,-allow_dup_skey,\
    -allow_tix,-requires_preauth,-requires_hwauth,-needchange,-allow_svr,\
    -password_changing_service,-support_desmd5 foo
    ```

    ```
    lsuser -R KRB5files -a krb5_attributes foo
    ```

  – A krb5_names módosításához és AIX felhasználónév/hosztnév pár hozzáadásához futtassa a következő parancsokat:

```
lsuser -R KRB5files -a krb5_names foo
chuser -R KRB5files krb5_names=bar:greenjeans.austin.ibm.com foo
lsuser -R KRB5files -a krb5_names foo
```

- ***Hogyan jeleníthetők meg a KRB5files fájlban megadott felhasználók?***

  Az összes Kerberos hitelesített felhasználó megjelenítéséhez adja ki a következő parancsot:

  ```
  lsuser -R KRB5files -a registry ALL
  ```

- ***Hogyan alakíthatok át egy AIX felhasználót Kerberos hitelesítésű felhasználóvá?***

  Az **mkseckrb5** parancs segítségével alakíthat át AIX felhasználót Kerberos hitelesített felhasználóvá. Az **mkseckrb5** parancs átalakítja a nem adminisztrátori felhasználókat (201-nél nagyobb felhasználói azonosítóval rendelkező felhasználók) Kerberos hitelesített felhasználókká. Az **mkseckrb5** parancs meghívásakor meg kell adnia a Hálózati hitelesítési szolgáltatás adminisztrátori azonosító nevét és jelszavát. Ha nem használja a véletlenszerű értékkel feltöltés lehetőséget, akkor az átalakítandó felhasználók jelszavát is meg kell adnia.

  Megjegyzés: Az **mkseckrb5** parancs csak a helyi felhasználókat alakítja át. A távoli tartományokban lévő felhasználók, mind például az LDAP, nem alakíthatók át ezzel a paranccsal.

  A következő példa *nem* használja a véletlenszerű értékkel feltöltés lehetőséget egy AIX felhasználónak Kerberos hitelesítésű felhasználóvá történő átalakítása során.

  1. Adja ki a következő parancsot:

     ```
     mkseckrb5 foo
     ```

  2. Mielőtt egy felhasználót bejelentkeztetne Kerberosszal, állítsa be a felhasználó SYSTEM és registry attribútumát az alábbiak szerint:

     ```
     chuser -R KRB5files SYSTEM=KRB5files registry=KRB5files foo
     ```

  A következő példa a véletlenszerű értékkel feltöltés lehetőséget használja egy AIX felhasználó Kerberos hitelesítésű felhasználóvá történő átalakítása során.

  1. Adja ki a következő parancsot:

     ```
     mkseckrb5 -r user1
     ```

  2. Az átalakítás befejezése után állítsa be a felhasználó SYSTEM és registry attribútumát, illetve a jelszavát az alábbiak szerint:

     ```
     chuser -R KRB5files SYSTEM=KRB5files registry=KRB5files user1
     passwd -R KRB5files user1
     ```

- ***Hogyan módosítható a Kerberos azonosítóhoz tartozó jelszó?***

  A root felhasználó beállíthatja a Kerberos azonosítóhoz tartozó jelszót a következő **passwd** parancs kiadásával:

  ```
  passwd -R KRB5files foo
  ```

  A következő üzenet jelenik meg a **passwd** parancs kiadása után:

  ```
  Changing password for "foo"
  foo's Old password:
  foo's New password:
  Enter the new password again:
  ```

  A **passwd** parancs root felhasználóként történő megadásakor a régi jelszó figyelmen kívül marad. A régi jelszó bekérését a methods.cfg fájl rootpwdrequired lehetőségének használatával letilthatja. A régi jelszó bekérésének letiltásához módosítsa az /usr/lib/security/methods.cfg fájlt a következőképp:

  ```
  KRB5files:
     options = db=BUILTIN,auth=KRB5,rootrequiresopw=false
  ```

- ***Hogyan kapható jegymegadási jegy sikeres bejelentkezés után, ha a needchange attribútum be van állítva?***

  Ha a needchange be van állítva, akkor a sikeres bejelentkezés után a **kinit** parancs meghívásával kérhet jegymegadási jegyet. A tárggyal kapcsolatos információkért tekintse meg a **needhange** attribútumot.

- ***Miért nem fogadja el a jelszavamat az AIX operációs rendszer?***

  Ha a jelszót nem fogadja el a rendszer, akkor tegye a következőket:

  – Ellenőrizze, hogy a KDC és a kadmind szerver fut-e.

– Ellenőrizze, hogy a jelszó megfelel-e az AIX operációs rendszer és a Hálózati hitelesítési szolgáltatás követelményeinek.

- ***Hogyan módosíthatók a jelszószabályok?***

  A jelszó szabályokat AIX operációs rendszeren a jelszóirányelv attribútumok módosításával változtathatja meg. A Hálózati hitelesítési szerver kadmin eszköz segítségével módosíthatja a Kerberos adatbázis jelszóirányelvét.

- ***Hitelesíthető egy Kerberos hitelesítésű felhasználó csak szabványos AIX hitelesítés által?***

  A Kerberos hitelesítésű felhasználó (foo) AIX **crypt()** hitelesítés használatával az alábbiak szerint hitelesíthető:

  1. Állítsa be a foo felhasználó AIX jelszavát (/etc/security/passwd) a **passwd** paranccsal.
  2. Válasszon másik jelszót tesztelés céljából. Például:

     ```
     passwd -R files foo
     ```

  3. Módosítsa a felhasználó SYSTEM attribútumát a következőképp:

     ```
     chuser -R KRB5files SYSTEM=compat foo
     ```

     A SYSTEM attribútum módosítása megváltoztatja a hitelesítési metódust Kerberosról **crypt()** metódusra.

     Megjegyzés: Mivel a példában lévő felhasználó helyi hitelesítéssel lép be, az AUTHSTATE értéke compat, és nem kerül kiadásra jegymegadási jegy. Ha a **crypt()** hitelesítést mentési mechanizmusként kívánja használni, akkor ugorjon a következő lépésre: 4.

  4. A **crypt()** hitelesítés mentési mechanizmusként történő használatához módosítsa a SYSTEM attribútumot az alábbiak szerint:

     ```
     chuser -R KRB5files SYSTEM="KRB5files or compat" foo
     ```

- ***Hogyan módosítható a kliens kadmind port?***

  A **kadmind** démon végrehajtja a Kerberos azonosítókezelést a Kerberos hitelesített rendszeren, amely NAS-t használ. A következő példa bemutatja a kliens **kadmind** port módosítását. Ebben a példában a **kadmind** démon kdcsrv.austin.ibm.com szerveren fut és a 812-es portot használja.

  1. A **config.krb5** parancs segítségével állítsa be a klienst:

     ```
     config.krb5 –C –r MYREALM –c kdcsrv.austin.ibm.com –s \
     kdcsrv.austin.ibm.com –d austin.ibm.com
     ```

  2. Szerkessze a krb5.conf fájlt és módosítsa a portszámot:

     ```
     admin_server = kdcsrv.austin.ibm.com:812
     ```

- ***Hogyan távolíthatók el a Kerberos hitelesítési adatok?***

  A Kerberos hitelesítési adatok a felhasználó minden bejelentkezésekor felülírásra kerülnek. Amikor a felhasználó kijelentkezik, ezek a hitelesítési adatok nem kerülnek eltávolításra. A hitelesítési adatok eltávolításához adja ki a következő NAS **kdestroy** parancsot:

  ```
  /usr/krb5/bin/kdestroy
  ```

- ***Hogyan módosítható a jegy életciklusa a KDC-n?***

  A jegy életciklusának módosításához KDC-n tegye a következőket:

  1. Módosítsa a max_life attribútumot a kdc.conf fájlban. Például:

     ```
     max_life = 8h 0m 0s
     ```

  2. Állítsa le, majd indítsa el a **krb5kdc** és **kadmind** démont.
  3. Módosítsa a krbtgt/MYREALM és kadmin/admin azonosító max_life értékét az 1. lépésben megadott értékre. Például:

     ```
     kadmin.local
     kadmin.local: modify_principal -maxlife "8 hours" krbtgt/MYREALM
     ```

- ***Mi történik, ha a kadmind démon nem elérhető?***

  Ha a kadmind démon nem érhető el, akkor a hitelesítés hosszabb ideig tarthat és meghiúsulhat. A hitelesítés meghiúsulhat, ha a hálózat azon része, ahol a kadmind démon található, nem érhető el vagy a kadmind szervert üzemeltető rendszer leállt. Ha a rendszer nem érhető el, akkor a kadmind lehetőség no értékre állítása a methods.cfg fájlban kiküszöböli a késleltetéseket a hitelesítés során.

Ha a kadmind démon nem fut, akkor a lejárt jelszóval rendelkező felhasználók nem tudnak bejelentkezni. Ha a kadmind démon nem érhető el (a démon nem fut vagy nem elérhető) és a felhasználó kiadja az **mkuser** parancsot, akkor a következő hiba jelenik meg:

```
3004-694 Error adding "krb5user": You do not have permission
```

Ezen felül a **chuser** és **lsuser** parancs csak az AIX-hez tartozó attribútumokat tudja kezelni, a Kerberoshoz tartozókat nem. A **rmuser** nem törli a Kerberos azonosítót, a **passwd** parancs pedig meghiúsul a Kerberos által hitelesített felhasználók esetében.

Ha a kadmind démon nem érhető el, akkor a root felhasználó nem tudja módosítani a felhasználói jelszavakat. Ha egy felhasználó elfelejti a jelszavát, akkor elérhetővé kell tenni a kadmind démont. Továbbá ha egy felhasználó a bejelentkezéskor Kerberos azonosítónevet ad meg, akkor az azonosítónév elsődleges nevét a rendszer megrövidíti (az AIX felhasználónév hosszkorlátozásának megfelelően). Ez a megrövidített név kerül felhasználásraaz AIX felhasználóazonosítási információk lekéréséhez (például a saját könyvtár érték lekéréséhez).

- ***Hogyan konfigurálhatom az AIX operációs rendszert Kerberos integrált bejelentkezésre LDAP AIX felhasználó/csoport kezeléssel?***

  Ha LDAP használatát terveziAIX felhasználó/csoport információk tárolására, akkor az **mksecldap** paranccsal végezze el az LDAP szerver és kliens beállítását, mielőtt futtatná az **mkkrb5srv** és **mkkrb5clnt** parancsokat. A Kerberos szerverek beállításához használja az **mkkrb5srv** parancsot. A Kerberos kliens beállításához használja az **mkkrb5clnt** parancsot –i LDAP kapcsolóval. Például:

  ```
  mkkrb5clnt -r MYREALM -c kdcsrv.ustin.ibm.com\
   –s kdcsrv.austin.ibm.com -a admin/admin -d austin.ibm.com -A -i LDAP -K –T
  ```

- ***Hogyan használhatók a Kerberosre felkészített távoli parancsok a sikeres bejelentkezés után?***

  Ha AIX felhasználó egy rendszerhez Kerberos hitelesítést használ, akkor a jegymegadási jegy használható Kerberosra felkészített távoli parancsokhoz.

  A következő példában a NAS szerver a kdcsrv.austin.ibm.com rendszeren az **mkkrb5srv** parancs segítségével van beállítva. Ez a rendszer Kerberos-alapú bejelentkezésekhez is be van állítva az **mkkrb5clnt** paranccsal. A második rendszer, a tx3d.austin.ibm.com, kliensként van beállítva az **mkkrb5clnt** paranccsal.

  1. Mentse el a host/tx3d.austin.ibm.com hosztazonosító kulcsait a tx3d rendszer /etc/krb5/krb5.keytab fájljába.
  2. Mivel az **mkkrb5clnt** parancs segítségével állította be a kliensgépet, ezek a kulcsok kibontásra kerültek a /var/krb5/security/keytab/tx3d.austin.ibm.com.keytab fájlba. Kösse a fájlt az /etc/krb5/krb5.keytab fájlhoz az alábbi módon:

     ```
     ln -s /var/krb5/security/keytab/tx3d.austin.ibm.com.keytab /etc/krb5/krb5.keytab
     ```

  3. Ha a tx3d.austin.ibm.com rendszer nem AIX Kerberos szerverrel van konfigurálva, akkor kifejezetten hozzon létre egy hosztazonosítót, és bontsa ki a kulcsokat. Például:

     ```
     kadmin -p admin/admin

     kadmin: addprinc -randkey host/tx3d.austin.ibm.com
     kadmin: ktadd -k /etc/krb5/krb5.keytab host/tx3d.austin.ibm.com
     kadmin:
     ```

     Mivel a kadmin eszköz a tx3d.austin.ibm.com rendszerről került meghívásra, a kulcsok a tx3d.austin.ibm.com rendszer /etc/krb5/krb5.keytab fájljába kerültek kibontásra. Ez a lépés a Kerberos admin szervert üzemeltető gépen hajtható végre (például a kdcsrv gépen). Miután kibontotta a kulcsokat a keytab fájlba, a fájl átvitelre és összefésülésre kerül a tx3d /etc/krb5/krb5.keytab fájljával.

  4. Engedélyezze a távoli parancsokat a Kerberos V5 hitelesítés használatához a tx3d.austin.ibm.com rendszeren:

     ```
     lsauthent
     Standard Aix
     chauthent -k5 -std
     lsauthent
     Kerberos 5
     Standard Aix
     ```

  5. Engedélyezze a távoli parancsokat Kerberos V5 hitelesítés használatához a kdcsrv.austin.ibm.com rendszeren:

     ```
     chauthent -k5 -std
     lsauthent
     Kerberos 5
     Standard Aix
     ```

6. Hozzon létre egy Kerberos hitelesített felhasználót (foo) a kdcsrv rendszeren és állítsa be a jelszót.

   ```
   mkuser -R KRB5files SYSTEM=KRB5files registry=KRB5files foo
   passwd -R KRB5files foo
   ```

7. Hozza létre a foo felhasználót a tx3d rendszeren:

   ```
   mkuser -R files foo
   ```

8. Jelentkezzen be telnet segítségével a kdcsrv.austin.ibm.com rendszerre Kerberos hitelesítéssel.
9. Annak ellenőrzéséhez, hogy jegymegadási jegy kiadásra került-e, adja ki a **klist** parancsot.

   ```
   /usr/krb5/bin/klist
   ```

A következőkben a Kerberosra felkészített távoli parancsokra látható példa:

Megjegyzés: Mielőtt futtatná a parancsokat a következő példákban, távolítsa el a .klogin, .rhost és hosts.equiv fájlt.

– Adja ki a **date** parancsot a távoli tx3d.austin.ibm.com hosztrendszeren az **rsh** paranccsal:

  ```
  rsh tx3d date
  ```

– Jelentkezzen be a távoli tx3d.austin.ibm.com rendszerre az **rlogin** paranccsal:

  ```
  hostname
  kdcsrv.austin.ibm.com
  rlogin tx3d -l foo
  ********************************
  * Welcome to AIX Version 6.1! *
  ********************************
  hostname
  tx3d.austin.ibm.com
  id
  uid=234(foo) gid=1(staff)
  ```

– Vigyen át egy fájlt a távoli tx3d.austin.ibm.com rendszerre az **rcp** paranccsal:

  ```
  rsh tx3d "ls -l /home/foo"
  total 0
  echo "Testing Kerberize-d rcp" >> xfile
  rcp xfile tx3d:/home/foo
  rsh tx3d "ls -l /home/foo"
  total 0
  -rw-r--r-- 1 foo staff 0 Apr 28 14:30 xfile
  rsh tx3d "more /home/foo/xfile"
  Testing Kerberize-d rcp
  ```

– Jelentkezzen be telnet segítségével a távoli tx3d.austin.ibm.com rendszerre Kerberos meghatalmazással:

  ```
  telnet tx3d
  Trying...
  Connected to tx3d.austin.ibm.com.
  Escape character is '^]'.
  [ Kerberos V5 accepts you as "foo@MYREALM" ]
  ```

– Jelentkezzen be telnet segítségével a tx3d.austin.ibm.com rendszerre, majd adja meg a hosztnevet és azonosítót, amikor a rendszer erre kéri:

  ```
  hostname
  tx3d.austin.ibm.com
  id
  uid=234(foo) gid=1(staff)
  ```

– A Kerberosra felkészített **ftp** parancs használata előtt a **kadmin** parancs (from tx3d.austin.ibm.com) segítségével hozza létre az ftp/tx3d.austin.ibm.com FTP szolgáltatásazonosítót és bontsa ki az /etc/krb5/krb5.keytab fájlba:

  ```
  kadmin: addprinc -randkey ftp/tx3d.austin.ibm.com@MYREALM
  kadmin: ktadd -k /etc/krb5/krb5.keytab ftp/tx3d.austin.ibm.com@MYREALM
  kadmin:
  ```

  A következőkben arra látható példa, hogy hogyan lehet bejelentkezni FTP-vel a távoli tx3d.austin.ibm.com rendszerre Kerberos meghatalmazással.

```
ftp tx3d
Name (tx3d:foo): foo
232 GSSAPI user foo@MYREALM is authorized as foo
230-Last login: Thu May 19 17:58:57 CDT 2005 on ftp from kdcsrv.austin.ibm.com
230 User foo logged in.
ftp> ftp> ls -la
```

Kerberos kliens beállítása nem AIX rendszeren futó Kerberos szerveren:

A AIX Kerberos kliens Kerberos szerverhez a következő nem AIX rendszerek esetén konfigurálható: Windows Active Directory, Solaris és HP.

Kerberos beállítása Windows Server Kerberos szolgáltatáshoz:

Számos módszer áll rendelkezésre a Kerberos beállításához Windows Server Kerberos szolgáltatáshoz.

A KRB5 Kerberos csak hitelesítés modulja használható az összetett betöltési modul hitelesítési részében. A konfiguráció során a felhasználó megadja a Kerberos környezetet a betöltési modulhoz. A KRB5 betöltési modulhoz a Kerberos használható alternatív módszerként a Windows 2000 vagy Windows 2003 Server Kerberos szolgáltatás hitelesítéséhez. Az AIX BUILTIN pszeudo betöltési modul hozzáférést biztosít a biztonsági függvénytár funkciókhoz. A BUILTIN betöltési modul kombinálható csak hitelesítés betöltési modulokkal az összetett betöltési modul adatbázisrészének biztosítása érdekében. Örökölt-felhasználó-és-csoport tároló és fájlrendszer elérést is biztosít. Az LDAP betöltési modul használható az összetett betöltési modul adatbázisrészeként.

A másik Kerberos környezet és AIX rendszer NAS esetétől eltérően ez a környezet nem biztosít Kerberos azonosítókezelést. A KRB5 betöltési modul olyan környezetben használható, ahol a Kerberos azonosítók nem AIX rendszeren vannak tárolva és nem kezelhetők az AIX operációs rendszerből a **kadmin** Kerberos adatbázis felülettel. A Kerberos azonosítókezelés külön történik, a Kerberos azonosítókezelési eszközeivel. Ezek az eszközök részei lehetnek szoftverszállítók által fejlesztett Kerberos termékeknek, vagy operációs rendszerbe, például a Windows 2000 rendszerbe lehetnek integrálva.

Windows Server 2000 Kerberos szolgáltatás beállítása:

A Windows Server 2000 Kerberos szolgáltatás és a NAS kliens együtt tud működni Kerberos protokoll szinten (RFC1510). Mivel a Windows Server 2000 nem támogatja a **kadmin** felületet, adja meg a –D kapcsolót az **mkkrb5clnt** paranccsal AIX kliensek beállítása során. Windows eszközök segítségével kezelje a Windows rendszeren lévő azonosítókat.

Tegye a következőket AIX kliens konfigurálásához Kerberos-alapú hitelesítéshez Windows Server 2000 Kerberos szolgáltatás esetén.

1. Állítsa be a Windows Server 2000 rendszert. Tekintse meg a Microsoft dokumentációt a Microsoft Active Directory Server beállításáért.
2. Ha a NAS kliens nincs telepítve az AIX kliensen, akkor telepítse a krb5.client.rte fájlkészletet az AIX bővítőcsomagból.
3. Használja az **mkkrb5clnt** parancsot az alábbi információkkal AIX Kerberos kliens konfigurálásához:

   realm Windows Active Directory Domain neve

   domain
   Az Active Directory szervert üzemeltető gép tartományneve

   KDC A Windows szerver hosztneve

   server A Windows szerver hosztneve

   Az alábbi lista az **mkkrb5clnt** parancs kimenetére mutat be egy példát:

   ```
   mkkrb5clnt -r MYREALM -d austin.ibm.com -c w2k.austin.ibm.com -s w2k.austin.ibm.com -D
   ```

A **mkkrb5clnt** parancs **-D** kapcsolója létrehozza az **is_kadmind_compat=no** paramétert az /etc/methods.cfg fájlban és beállítja a Kerberos kliens környezetet a nem AIX rendszerek hitelesítéséhez. Ne használja a **-D** paramétert az **mkkrb5clnt** paranccsal a Kerberos klienskörnyezet beállításához IBM Hálózati hitelesítési szolgáltatás (NAS) esetén.

Megjegyzés: Az **mkkrb5clnt** parancs futtatásakor a következő szakasz kerül a methods.cfg fájlba.

```
KRB5:
        program = /usr/lib/security/KRB5
        program_64 = /usr/lib/security/KRB5_64
        options = authonly,is_kadmind_compat=no

KRB5files:
        options = db=BUILTIN,auth=KRB5
```

További információkat

- az **mkkrb5clnt** parancsról és a megengedett kapcsolókról itt talál: **mkkrb5clnt** parancs.
- a methods.cfg fájlról itt talál: methods.cfg fájl.

4. Mivel a Windows támogatja a DES-CBC-MD5 és DES-CBC-CRC titkosítási típust, módosítsa a krb5.conf fájlinformációkat a következőhöz hasonlóra:

   ```
   [libdefaults]
       default_realm = MYREALM
       default_keytab_name = FILE:/etc/krb5/krb5.keytab
       default_tkt_enctypes = des-cbc-md5 des-cbc-crc
       default_tgs_enctypes = des-cbc-md5 des-cbc-crc
   ```

5. Hozzon létre egy hosztazonosítót.

   Mivel a Windows fióknevek nem rendelkeznek több résszel, mint a NAS azonosítónevek, nem hozható létre közvetlenül fiók a teljes képzésű hosztnév segítségével (host/<*teljes_képzésű_hosztnév*>). Az azonosítópéldány ehelyett a szolgáltatás-azonosító-név leképezéssel kerül létrehozásra. Ebben az esetben a hosztazonosítónak megfelelő fiók létrejön, és az azonosító-név leképezés hozzáadásra kerül.

   Az Active Directory szerveren használja az Active Directory felhasználókezelési eszközt olyan új felhasználói fiók létrehozásához, amely megfelel a tx3d.austin.ibm.com AIX kliensnekaz alábbiak szerint:

   a. Válassza ki a Felhasználó mappát.
   b. Kattintson a jobb egérgombbal az Új lehetőség kiválasztásához.
   c. Válassza ki a Felhasználó lehetőséget.
   d. Adja meg a tx3d értéket az Keresztnév mezőben, majd kattintson a Tovább gombra.
   e. Hozzon létre egy jelszót, majd kattintson a Tovább gombra.
   f. Kattintson a Befejezés gombra a hosztazonosító létrehozásához.

6. A Windows Server 2000 számítógépen adja meg parancssorból a **Ktpass** parancsot a tx3d.keytab fájl létrehozásához, és állítson be egy AIX hoszt fiókot az alábbiak szerint:

   ```
   Ktpass -princ host/tx3d.austin.ibm.com@MYREALM -mapuser tx3d -pass password -out tx3d.keytab
   ```

7. Másolja át a tx3d.keytab fájlt az AIX hosztrendszerre.
8. Fésülje össze a tx3d.keytab fájlt az /etc/krb5/krb5.keytab fájllal azAIX rendszeren az alábbiak szerint:

   ```
   ktutil
   rkt tx3d.keytab
   wkt /etc/krb5/krb5.keytab
   q
   ```

9. Hozza létre a Windows tartományi fiókokat az Active Directory felhasználókezelési eszközeivel.
10. Olyan AIX fiókok létrehozásához, amelyek megfelelnek a Windows-tartomány fiókjainak és Kerberos hitelesítést használnak, futtassa a következő parancsot:

    ```
    mkuser registry=KRB5files SYSTEM=KRB5files foo
    ```

11. Jelentkezzen be az AIX rendszerre, és a konfiguráció ellenőrzéséhez futtassa a **telnet** parancsot.

    Az alábbiakban egy olyan Kerberos integrált bejelentkezési munkamenet példája látható, amely KRB5 használatát valósítja meg Windows Active Directory esetén:

```
telnet tx3d

Trying...
Connected to tx3d.austin.ibm.com.
Escape character is '^]'.

telnet (tx3d.austin.ibm.com)
login: foo
foo's Password:
*******************************************************************************
* Welcome to AIX Version 6.1! *
*******************************************************************************
echo $AUTHSTATE
KRB5files

/usr/krb5/bin/klist
Ticket cache: FILE:/var/krb5/security/creds/krb5cc_foo@AUSTIN.IBM.COM_203
Default principal: foo@AUSTIN.IBM.COM

Valid starting Expires Service principal
04/29/05 14:37:28 04/30/05 00:39:22 krbtgt/AUSTIN.IBM.COM@AUSTIN.IBM.COM
     Renew until 04/30/05 14:37:28

04/29/05 14:39:22 04/30/05 00:39:22 host/tx3d.austin.ibm.com@AUSTIN.IBM.COM
```

Windows Server 2003 Kerberos szolgáltatás beállítása:

A Kerberos kliens beállítható Windows Server 2003 Kerberos szolgáltatáshoz.

Windows Server 2003 Kerberos szolgáltatáshoz AIX kliens beállítása érdekében kövesse a következő lépéseket: “Windows Server 2000 Kerberos szolgáltatás beállítása” oldalszám: 299.

Megjegyzés: A NAS **kpasswd** kliens segédprogram nem tudja módosítani a Kerberos azonosító jelszavát Windows Server 2003 Kerberos szolgáltatáson. Ezért a Kerberost használó AIX rendszerre történő sikeres bejelentkezést követően a felhasználó nem tudja megváltoztatni a jelszót a Windows Server 2003 rendszeren.

Kerberos beállítása Sun Solaris és HP-UX Kerberos tartományvezérlőkhöz:

A Kerberos kliens beállítható Sun Solaris és HP-UX Kerberos tartományvezérlőkhöz.

A Kerberos környezet és AIX rendszer NAS esetétől eltérően ez a környezet nem biztosít Kerberos azonosítókezelést. A KRB5 betöltési modul olyan környezetben használható, ahol a Kerberos azonosítók nem AIX rendszeren vannak tárolva és nem kezelhetők az AIX operációs rendszerből a **kadmin** Kerberos adatbázis felülettel. A Kerberos azonosítókezelés külön történik, a Kerberos azonosítókezelési eszközeivel. Ezek az eszközök lehetnek a szoftverszállítók által fejlesztett Kerberos termék részei vagy integrálhatók az operációs rendszerbe.

Sun Solaris beállítása:

A Kerberos kliens beállítható Sun Solaris rendszeren.

A Sun Enterprise Authentication Mechanism (SEAM) és az AIX NAS kliens együttműködésre képes Kerberos protokollszinten (RFC1510). Mivel a Solaris **kadmind** démon felület nem kompatibilis azAIX NAS kliens **kadmin** felületével, adja meg a –D kapcsolót az **mkkrb5clnt** paranccsal AIX kliensek konfigurálása során. Solaris eszközök segítségével végezzen azonosítókezelést a Solaris rendszereken. Mivel a jelszóváltoztatási protokoll eltérő a SEAM Kerberos szerverek és az AIX NAS kliensek esetén, egy azonosító jelszavának megváltoztatása miatt a konfiguráció meghiúsul.

A Solaris kerül felhasználásra a következő példában.

A következőkkel állítson be egy AIX klienset Kerberos-alapú hitelesítésre SEAM esetén.

1. Állítsa be a SEAM-ot a Sun dokumentáció segítségével.
2. Ha a NAS kliens nincs telepítve az AIX kliensen, akkor telepítse a krb5.client.rte fájlkészletet az AIX bővítőcsomagból.
3. AIX Kerberos kliens konfigurálásához használja az **mkkrb5clnt** parancsot az alábbi konfigurációs információkkal:

   tartomány
   Solaris Kerberos tartomány neve: AUSTIN.IBM.COM

   domain
   A Kerberos szervereket üzemeltető gép tartományneve: Austin.ibm.com

   KDC A KDC-t üzemeletető Solaris rendszer hosztneve: sunsys.austin.ibm.com

   server A **kadmin** démont üzemeltető Solaris rendszer hosztneve (általában ugyanaz, mint a KDC esetén): sunsys.austin.ibm.com

   Megjegyzés: Mivel a Solaris és AIX NAS kliens **kadmin** felülete eltérő, a szerver nevét nem használják a NAS kliensek, és a –D kapcsolót kell használnia az **mkkrb5clnt** paranccsal.

   Az alábbi lista az **mkkrb5clnt** parancs kimenetére mutat be egy példát:

   ```
   mkkrb5clnt -r AUSTIN.IBM.COM -d austin.ibm.com\
    -c sunsys.austin.ibm.com -s sunsys.austin.ibm.com -D
   ```

   A **-D** paraméter az **mkkrb5clnt** parancs esetén létrehozza az**is_kadmind_compat=no** beállítást az /etc/security/methods.cfg fájlban, és a Kerberos klienskörnyezetet nem AIX rendszerek használatára állítja be. Ne használja a **-D** paramétert az **mkkrb5clnt** paranccsal a Kerberos klienskörnyezet beállításához IBM Hálózati hitelesítési szolgáltatás (NAS) esetén.

   Megjegyzés: Az **mkkrb5clnt** parancs futtatásakor a következő szakasz kerül a methods.cfg fájlba.

   ```
   KRB5:
           program = /usr/lib/security/KRB5
           program_64 = /usr/lib/security/KRB5_64
           options = authonly,is_kadmind_compat=no

   KRB5files:
           options = db=BUILTIN,auth=KRB5
   ```

   További információkat

   - az **mkkrb5clnt** parancsról és a megengedett kapcsolókról itt talál: **mkkrb5clnt** parancs.
   - a methods.cfg fájlról itt talál: methods.cfg fájl.
4. A Solaris **kadmin** eszköz segítségével hozzon létre egy host/tx3d.austin.ibm.com@MYREALM hosztazonosítót és mentse el azt egy fájlba, a következőhöz hasonlóan:

   ```
   kadmin: add_principal -randkey host/tx3d.austin.ibm.com
   Principal "host/tx3d.austin.ibm.com@AUSTIN.IBM.COM" created.

   kadmin:ktadd -k /tmp/tx3d.keytab host/tx3d.austin.ibm.com
   Entry for principal host/tx3d.austin.ibm.com with kvno 3,
       encryption type DES-CBC-CRC added to keytab WRFILE:/tmp/tx3d.keytab.

   kadmin: quit
   ```
5. Másolja át a tx3d.keytab fájlt az AIX hosztrendszerre.
6. Fésülje össze a tx3d.keytab fájlt az /etc/krb5/krb5.keytab fájllal azAIX rendszeren az alábbiak szerint:

   ```
   ktutil
   rkt tx3d.keytab
   l
   slot   KVNO   Principal
   wkt /etc/krb5/krb5.keytab
   q
   ```
7. Kerberos azonosító létrehozásához használja a Solaris **kadmin** eszközt.

   ```
   add_principal sunuser
   ```

8. Olyan AIX fiókok létrehozásához, amelyek megfelelnek a Solaris Kerberos azonosítónak, és Kerberos hitelesítést használnak, adja meg a következő parancsot:

   ```
   mkuser registry=KRB5files SYSTEM=KRB5files sunuser
   ```

9. A **telnet** paranccsal jelentkezzen be az AIX rendszerre a sunuser felhasználónévvel és jelszóval, majd ellenőrizze a konfigurációt.

   A következő egy Kerberos integrált bejelentkezési munkamenet, amely KRB5-öt használ Solaris KDC rendszeren:

   ```
   telnet tx3d

   echo $AUTHSTATE
   KRB5files

   echo $KRB5CCNAME
   FILE:/var/krb5/security/creds/krb5cc_sunuser@AUSTIN.IBM.COM_207

   View credentials:
   /usr/krb5/bin/klist
   ```

HP-UX beállítása:

A Kerberos kliens beállítható HP-UX rendszeren.

A HP-UX 11i hitelesítési lépései a “Sun Solaris beállítása” oldalszám: 301 lépéseihez hasonlók. A HP-UX KDC és AIX NAS kliens együttműködésre képes a Kerberos protokollszintjén (RFC1510). A jelszó-módosítási protokoll szintén kompatibilis. Mivel a HP-UX **kadmind** démon felület nem kompatibilis az AIX NAS kliens **kadmin** felületével, meg kell adnia a –D kapcsolót az**mkkrb5clnt** parancshoz AIX kliensek konfigurálása során.

A következőkkel konfigurálhat AIX klienset Kerberos alapú hitelesítéshez HP-UX 11i Kerberos v2.1 esetén.

1. HP-UX 11i Kerberos V2.1 beállítása a HP dokumentáció segítségével.
2. Ha a NAS kliens nincs telepítve az AIX kliensen, akkor telepítse a krb5.client.rte fájlkészletet az AIX bővítőcsomagból.
3. Használja az **mkkrb5clnt** parancsot az alábbi információkkal AIX Kerberos kliens konfigurálásához:

   tartomány
   HP Kerberos tartomány neve: HPSYS.AUSTIN.IBM.COM

   domain
   A gép tartományneve, amely a HP-UX Kerberos szervereket szolgálja ki: austin.ibm.com

   KDC A HP-UX rendszer hosztneve, amely a KDC-t szolgálja ki: hpsys.austin.ibm.com

   server A HP-UX szerver neve: hpsys.austin.ibm.com

   Megjegyzés: Mivel a HP-UX és AIX NAS kliens**kadmin** felületei eltérőek, a szerver nevét nem használják a NAS kliensek, és a –D kell használni az **mkkrb5clnt** paranccsal.

   Az alábbi lista az **mkkrb5clnt** parancs kimenetére mutat be egy példát:

   ```
   mkkrb5clnt -r AUSTIN.IBM.COM -d austin.ibm.com\
    -c hpsys.austin.ibm.com -s hpsys.austin.ibm.com -D
   ```

   A **-D** paraméter az **mkkrb5clnt** parancs esetén létrehozza az**is_kadmind_compat=no** beállítást az /etc/security/methods.cfg fájlban, és a Kerberos kliens környezetet nem AIX rendszerek használatára állítja be. Ne használja a **-D** paramétert az **mkkrb5clnt** paranccsal a Kerberos kliens környezet beállításához IBM Hálózati hitelesítési szolgáltatás (NAS) esetén.

   Megjegyzés: Az **mkkrb5clnt** parancs futtatásakor a következő szakasz kerül a methods.cfg fájlba.

   ```
   KRB5:
           program = /usr/lib/security/KRB5
           program_64 = /usr/lib/security/KRB5_64
   ```

```
        options = authonly,is_kadmind_compat=no

KRB5files:
        options = db=BUILTIN,auth=KRB5
```

További információkat

- az **mkkrb5clnt** parancsról és a megengedett kapcsolókról itt talál: **mkkrb5clnt** parancs.
- a methods.cfg fájlról itt talál: methods.cfg fájl.

4. Módosítsa a krb5.conf fájlt, hogy a titkosítási típus megfeleljen a HP-UX Kerberos beállítás (**krbsetup**) során használt értéknek. Ha DES-CRC érték kerül felhasználásra, akkor a [libdefaults] szakaszt a krb5.conf fájlban az AIX kliensen az alábbiak szerint szerkessze:

   ```
   default_tkt_enctypes = des-cbc-crc

   default_tgs_enctypes = des-cbc-crc
   ```

5. A HP-UX **kadmin_ui** eszköz segítségével létrehozhat egy host/tx3d.austin.ibm.com hosztazonosítót.
6. Bontsa ki a kulcsot és mentse el egy fájlba. Az Azonosítóinformációk ablak Szerkesztés menüjében válassza ki a Szolgáltatáskulcs kibontása lehetőséget a kulcsok kibontásához.
7. Másolja át a tx3d.keytab fájlt az AIX hosztrendszerre.
8. Fésülje össze a tx3d.keytab fájlt az /etc/krb5/krb5.keytab fájllal azAIX rendszeren az alábbiak szerint:

   ```
   ktutil
   rkt tx3d.keytab
   l
   slot   KVNO   Principal
   wkt /etc/krb5/krb5.keytab
   q
   ```

9. A HP-UX **kadmin_ui** eszköz segítségével hozzon létre egy hpuser Kerberos azonosítót, majd kattintson a Szerkesztés/attribútum lapra a pw_require kapcsoló törléséhez.
10. Hozzon létre egy AIX fiókot, amely megfelel a Kerberos azonosítónak HP-UX rendszeren, az alábbiak szerint:

    ```
    mkuser registry=KRB5files SYSTEM=KRB5files hpuser
    ```

11. A **telnet** paranccsal jelentkezzen be az AIX rendszerre a hpuser felhasználónévvel és jelszóval, majd ellenőrizze a konfigurációt.

    A következő egy Kerberos integrált bejelentkezési munkamenet, amely KRB5-öt használ HP-UX rendszeren:

    ```
    telnet tx3d

    echo $AUTHSTATE
    KRB5files

    View credentials:
    /usr/krb5/bin/klist
    ```

12. A **passwd** parancs segítségével módosítsa a jelszót.

    Megjegyzés: A HP-UX jelszóirányelv betartatásra kerül a jelszó módosítása közben. A jelszóirányelv beállításával kapcsolatos részleteket a HP-UX dokumentáció tartalmaz.

Kerberos nem AIX rendszereken: kérdések és hibaelhárítási információk:

Ez azon Kerberos kliensekkel kapcsolatos kérdésekre ad választ, amelyek Kerberos szervert használnak nem AIX rendszereken.

Megjegyzés: A Microsoft Active Directory Server kerül alkalmazásra a következő példákban. Azonban ezek a példák alkalmazhatók a Solaris és HP rendszerekre.

A hibaelhárítás első lépéseként győződjön meg róla, hogy az összes szerver és démon fut.

A Kerberos nem AIX rendszereken a syslog alrendszert használja a hibákkal és hibakereséssel kapcsolatos információk írásához. A syslog naplózással kapcsolatos információkat a **syslogd** démon tartalmaz.

- ***Hogyan hozható létre AIX felhasználó?***

  Hozzon létre egy AIX felhasználói fiókot (foo) a következő parancs futtatásával:

  ```
  mkuser registry=KRB5files SYSTEM=KRB5files foo
  ```

  Azz **mkuser** parancs létrehoz egy felhasználót az AIX rendszeren. A felhasználó számára létre kell hozni a Windows Server Active Directory-n egy AIX fióknak megfelelő fiókot. A felhasználói fiók létrehozása Windows Server Active Directory-n implicit módon létrehozza az azonosítókat.

- ***Hogyan távolítható el a Kerberos hitelesített felhasználó?***

  Kerberos hitelesített felhasználó eltávolításához futtassa a következő parancsot:

  ```
  rmuser –R KRB5files foo
  ```

  Az **rmuser** parancs eltávolít egy felhasználót az AIX rendszerről. A felhasználót a Windows Server Active Directory-ból is el kell távolítani a Windows Server felhasználókezelési eszközökkel.

- ***Hogyan módosítható egy Kerberos hitelesített felhasználó jelszava?***

  Kerberos hitelesített felhasználó jelszavának módosításához futtassa a következő parancsot:

  ```
  passwd –R KRB5files foo
  ```

  Ha a KDC támogatja a **kpasswd** parancsot, akkor a **passwd** parancs megváltoztatja a foo@MYREALM Kerberos azonosítót jelszavát a Kerberos szerveren.

- ***Hogyan engedélyezhető, hogy a felhasználók módosíthassák a lejárt jelszavukat a kliensen?***

  Ahhoz, hogy a felhasználók módosíthassák jelszavukat a kliensen, adja hozzá az allow_expired_pwd=yes lehetőséget a methods.cfg fájlhoz. Ha a lehetőség értéke yes, akkor a lejárt jelszóval rendelkező felhasználóknak módosítaniuk kell a lejárt jelszavukat. Ha a lehetőség értéke no vagy not present, akkor a felhasználók nem hitelesíthetők.

  ```
  KRB5:
     program = /usr/lib/security/KRB5
     options = authonly,allow_expired_pwd=yes
  ```

- ***Hogyan alakítható át az AIX felhasználó Kerberos hitelesített felhasználóvá?***

  AIX felhasználó Kerberos hitelesített felhasználóvá alakításához tegye a következőket:

  1. A következő parancs futtatásával ellenőrizze, hogy a felhasználó rendelkezik-e fiókkal a Windows Server Active Directory-n:

     ```
     chuser registry=KRB5files SYSTEM=KRB5files foo
     ```

  2. Ha a felhasználó nem rendelkezik fiókkal az Active Directory-n, akkor hozzon létre egy fiókot az Active Directory-n és állítsa be a SYSTEM és registry attribútumot a **chuser** parancs segítségével. Elképzelhető, hogy az Active Directory fiók felhasználóneve nem egyezik meg az AIX felhasználónévvel. Ha más nevet használ AIX felhasználónévként, akkor az auth_name attribútum segítségével képezze azt le Active Directory névre.

     ```
     chuser registry=KRB5files SYSTEM=KRB5files auth_name=Christopher chris
     ```

- ***Mi a teendő, ha elfelejtettem a jelszót?***

  A jelszó elfelejtése esetén a jelszót az Active Directory adminisztrátornak módosítania kell. Az AIX root felhasználó nem tudja beállítani az Active Directory Kerberos azonosító jelszavát.

- ***Mi az auth_name és auth_domain attribútumok célja?***

  Megjegyzés: Ezek az attribútumok elhagyhatóak. Ha az AIX rendszer támogatja a nyolc karakternél hosszabb felhasználóneveket, akkor elképzelhető, hogy az auth_name attribútumra nincs szükség.

  Az auth_name és az auth_domain attribútum leképezi az AIX felhasználóneveket Kerberos azonosítónevekre a KDC-n. Ha például a chris AIX felhasználó auth_name=christopher és auth_domain=SOMEREALM attribútummal rendelkezik, akkor a Kerberos azonosítónév christopher@SOMEREALM lesz. Az auth_domain attribútum használatával a kérések a SOMEREALM nevű tartományhoz kerülnek elküldésre az alapértelmezett tartománynév helyett. Ezáltal a chris felhasználó a MYREALM tartomány helyett a SOMEREALM tartományban hitelesítheti magát. Ebben a példában a krb5.conf fájlt szintén módosítani kell a SOMEREALM tartománynév megadása érdekében.

- ***Kerberos hitelesített felhasználó hitelesíthető szabványos AIX hitelesítéssel?***

  Igen, a Kerberos által hitelesített felhasználó hitelesíthető szabványos AIX hitelesítéssel a következőképp:

1. Állítsa be az AIX jelszót (/etc/security/passwd) a **passwd** parancs segítségével:
   ```
   passwd -R files foo
   ```
2. Módosítsa a felhasználó registry és SYSTEM attribútumát a következőképp:
   ```
   chuser -R KRB5files registry=files SYSTEM=compat foo
   ```
   Ez a parancs a Kerberos hitelesítést compat hitelesítésre módosítja (amely a crypt szubrutint használja). A foo felhasználó következő bejelentkezésekor az /etc/security/passwd fájlból származó helyi jelszó kerül felhasználásra.

   A crypt hitelesítés mentési mechanizmusként is használható a SYSTEM attribútum módosításával, hogy engedélyezze a helyi hitelesítést, amennyiben a Kerberos hitelesítés sikertelen, a következőképp:
   ```
   chuser -R KRB5files SYSTEM="KRB5files or compat" foo
   ```

- ***Be kell állítani Kerberos szervert az AIX rendszeren Windows Server 2000 Kerberos szolgáltatás használata esetén?***

  Nem kell beállítani Kerberos szervert (KDC) az AIX kliensen, mivel a felhasználók az Active Directory KDC-n végzik a hitelesítést. Ha az AIX Hálózati hitelesítési szolgáltatás KDC-t kívánja Kerberos szerverként használni bizonyos célra, akkor a Kerberos szervert be kell állítani.

- ***Mi a teendő, ha az AIX nem fogadja el a jelszót?***

  Ha az AIX nem fogadja el a jelszót, akkor tegye a következőket:
  - Győződjön meg róla, hogy a kliens kommunikál a Windows 2000 Active Directory Serverrel
  - Győződjön meg róla, hogy a jelszó az AIX és Windows Server 2000 Active Directory rendszernek egyaránt megfelel. Az AIX jelszóirányelv-szabályainak módosításával kapcsolatos információkat a Megjelenítési irányelv módosítása rész tartalmaz.

  Megjegyzés: A Windows Server 2003 Kerberos szolgáltatás jelszava nem módosítható.

- ***Mi a teendő, ha nem tudok bejelentkezni a rendszerre?***

  Ha nem tud bejelentkezni a rendszerre, akkor tegye a következőket:
  - A következőképp ellenőrizze a Windows rendszeren, hogy a KDC fut-e:
    1. A Vezérlőpulton válassza ki a Felügyeleti eszközök ikont.
    2. Válassza ki a Szolgáltatások ikont.
    3. Győződjön meg róla, hogy a Kerberos kulcselosztó központ elindult.
  - AIX rendszeren ellenőrizze, hogy az /etc/krb5/krb5.conf fájl a megfelelő KDC-re mutat, és hogy érvényes paraméterekkel rendelkezik.
  - AIX rendszeren ellenőrizze, hogy a client-keytab fájl tartalmazza-e a hosztkulcsot. Ha például az alapértelmezett keytab fájl az /etc/krb5/krb5.keytab, akkor futtassa a következőt:
    ```
    ktutil
    rkt /etc/krb5/krb5.keytab
    l
    ```
  - Ellenőrizze, hogy a **kvno** parancs kimenete a keytab fájlban megfelelő a **Ktpass** parancs kimenetének.
  - Győződjön meg róla, hogy az auth_name és auth_domain attribútum be van állítva, és érvényes azonosítót határoz meg az Active Directory kulcselosztó központban.
  - Ellenőrizze, hogy a SYSTEM attribútum Kerberos bejelentkezéshez van-e beállítva.
  - Ellenőrizze, hogy a jelszó nem járt-e le.

- ***Hogyan tiltható le a jegymegadási jegy ellenőrzés?***

  A jegymegadási jegy ellenőrzés az /usr/lib/security/methods.cfg fájl KRB5 szakaszában megadott bejegyzéssel tiltható le a következőképp:
  ```
  KRB5:
     program = /usr/lib/security/KRB5
     options = tgt_verify=no
  KRB5files:
     options = db=BUILTIN,auth=KRB5
  ```

A tgt_verify lehetséges értéke: no és false a jegymegadási jegy ellenőrzés letiltásához, illetve yes és true a jegymegadási jegy ellenőrzésének engedélyezéséhez. A jegymegadási jegy ellenőrzés alapértelmezésben engedélyezett. Ha a tgt_verify lehetőséget no értékre állítja, akkor a jegymegadási jegy ellenőrzés le van tiltva, és nem kell átvinni a hosztazonosító kulcsokat. A módosítás esetén a keytab fájlra nem lesz szükség a hitelesítéshez. Más Kerberosra felkészített alkalmazások számára szükség lehet a keytab fájlra a hoszt- és szolgáltatásazonosítókhoz.

- ***Mi a teendő, ha nem lehet bejelentkezni, mivel a hosztnév nem kerül feloldásra, a teljes képzésű hosztnév használata pedig sikertelen?***

  A jegymegadási jegy ellenőrzéséhez egy host/<hosztnév> azonosítót létre kell hozni a kulcselosztó központban. A hosztnév a kliens teljes képzésű neve, amelyen a hitelesítés történik. A kliensrendszer szolgáltatásjegyet kér a host/<hosztnév> hosztazonosító névvel. Bizonyos konfigurációkban a kliensgép nem tudja lekérni a teljes képzésű hosztnevet, hanem ehelyett egy rövid nevet kér. Az ilyen példányokban eltérési hiba történik, a példány jegymegadási jegy ellenőrzése és a bejelentkezés pedig meghiúsul. Ha például az /etc/hosts csak rövid névvel rendelkezik és az /etc/netsvc.conf fájl megadja a hosts=local,bind értéket, akkor a névfeloldás visszaadja a rövid nevet.

  A névfeloldási probléma kijavítása érdekében tegye a következőt:

  – Módosítsa a névfeloldási sorrendet az /etc/netsvc.conf fájlban, hogy a teljes képzésű hosztnév kerüljön visszaadásra. A netsvc.conf fájl megadja a hosztnevek és álnevek feloldási sorrendjét.

    A következő példában a feloldó a BIND szolgáltatást használja a hosztnév feloldásához. Ha a BIND szolgáltatás sikertelen, akkor a feloldó az /etc/hosts fájlt használja. Ha mindkét metódus meghiúsul, akkor a feloldó a nist használja.

    ```
    hosts=bind,local,nis
    ```

    Ha a keresési sorrendben használt első módszernek localnak kell lennie, akkor módosítsa a rövid nevet (myhost) az /etc/hosts fájlban a teljes képzésű hosztnévre (myhost.austin.ibm.com).

  – Ha a jegymegadási jegy ellenőrzésre nincs szükség, akkor a *Hogyan tiltható le a jegymegadási jegy ellenőrzés?* részben megtalálja a jegymegadási jegy ellenőrzés letiltására szolgáló útmutatást.

- ***Miért ad vissza 0-t a passwdexpired szubrutin, amikor a Kerberos felhasználói jelszó lejárt a nem AIX Kerberos szerveren?***

  A **passwdexpired** szubrutin azért ad vissza 0-t, mert a jelszólejárati információk a **kadmin** felületek inkompatibilitása vagy elérhetetlensége miatt nem szerezhetők meg közvetlenül a nem AIX Kerberos szerverről.

  A **methods.cfg** fájlban található **allow_expired_pwd** kapcsoló lehetővé teszi az AIX számára a jelszólejárati információk megszerzését a Kerberos hitelesítési felületek segítségével. A jelszólejárati információk aktuális állapotát vagy a bejelentkezés során, vagy az **authenticate** szubrutin és a **passwdexpired** szubrutin meghívásával lehet lekérdezni.

Kerberos modul

A Kerberos modul az NFS kliens és szerver kód által használ kernel kiterjesztés. Lehetővé teszi az NFS kliens és szerver kód számára a Kerberos üzenetek integritás és adatvédelmi függvényeinek végrehajtását a **gss** démonhoz küldött hívások nélkül.

A Kerberos modult a **gss** démon tölti be. A használt metódusok a Network Authentication Service 1.2 változatától függenek, aminek viszont az MIT Kerberos az alapja.

A Kerberos modul helye: /usr/lib/drivers/krb5.ext.

Kapcsolódó információkért tekintse meg a **gss** démont.

Kapcsolódó tájékoztatás:

IBM developerWorks erőforrások az IBM Hálózati hitelesítési szolgáltatáson és kapcsolódó technológiákon AIX rendszerhez

Távoli hitelesítés behívásos felhasználói szolgáltatás szervere

Az IBM Távoli hitelesítés behívásos felhasználói szolgáltatása (RADIUS) egy hitelesítésre, felhatalmazásra és számlázásra tervezett hálózati elérési protokoll. A portalapú protokoll a Hálózatelérési szerverek (NAS) és a hitelesítő és könyvelő szerverek közötti kommunikációt határozza meg.

Egy NAS szerverek RADIUS kliensekként működnek. A kliens és a RADIUS szerver közötti tranzakciók hitelesítése egy *osztott titkon* keresztül történik, amelyet a felek nem küldenek át a hálózaton. A kliens és a RADIUS szerver között küldött felhasználói jelszavak titkosítva vannak.

A kliens felelőssége a felhasználói információk átadása a kijelölt RADIUS szervernek, majd a kapott válasznak megfelelően a tevékenység folytatása. A RADIUS szerver felelőssége a felhasználó csatlakozási kérésének fogadása, majd az összes olyan konfigurációs információ visszaadása a kliensnek, amelyek alapján az képes a kívánt szolgáltatást biztosítani a felhasználó számára. A RADIUS szerver fejlett proxybeállítások konfigurálása esetén működhet más RADIUS szerverek proxy klienseként. A RADIUS Felhasználói adatcsomag protokollt (**UDP**) használ szállítási protokollként.

A RADIUS hitelesítési és feljogosítási protokollja az IETF RFC 2865 szabványon alapul. A szerver az RFC 2866-ban meghatározott könyvelési protokollt is biztosítja. A további támogatott szabványok: RFC 2284 (EAP), az RFC 2869 egyes részei, valamint az RFC 2882, MD5-Challenge és TLS jelszó lejáratra vonatkozó üzenetei. Ezekről az RFC-kről az alábbi helyeken talál információkat:

IETF RFC 2865
http://www.ietf.org/rfc/rfc2865.txt

RFC 2866
http://www.ietf.org/rfc/rfc2866.txt

RFC 2284
http://www.ietf.org/rfc/rfc2284.txt

RFC 2869
http://www.ietf.org/rfc/rfc2869.txt

RFC 2882
http://www.ietf.org/rfc/rfc2882.txt

Az összes RFC szabványt megtalálja a http://www.ietf.org Internet címen.

RADIUS szerver telepítése

A RADIUS szervert a Rendszerfelügyeleti illesztő eszközzel vagy a **installp** paranccsal telepítheti. A RADIUS szoftver az AIX alap adathordozóján található, a képfájl nevek: radius.base és bos.msg.<lang>4.rte.

Ha a felhasználóneveket és jelszavakat LDAP címtárban kívánja tárolni, akkor az ldap.server csomagot is telepítenie kell. Az **installp** szoftvert telepíteni kell minden olyan kliensen, ahol a RADIUS telepítve van.

Ha használni kívánja az EAP-TLS hitelesítést (például a digitális tanúsítványok hitelesítéséhez a vezetéknélküli hálózaton), az OpenSSL 0.9.7 vagy újabb változatot is telepíteni kell, és meg kell adni a libssl.a függvénytár teljes elérési útját az /etc/radius/radiusd.conf konfigurációs fájlban.

A RADIUS démonok a **radiusctl** paranccsal indíthatók el. Elindításuk után több radiusd folyamat fut, a következőkhöz egy-egy:

- felhatalmazás
- számlázás
- más démonok megfigyelése

Újraindításkor a démonok automatikusan elindulnak a 2-es futási szinten, hacsak a RADIUS nincs beállítva EAP-TLS-hez.

A rutin megváltoztatásához módosítsa az /etc/rc.d/rc2.d/Sradiusd fájlt.

Megjegyzés: Ha a RADIUS be van állítva a digitális tanúsítványok hitelesítésére EAP-TLS-sel, akkor a démonok nem állíthatók be automatikus indulásra, mivel a tanúsítvány jelmondatot az adminisztrátornak kell megadnia, amely a RADIUS kézi indítását és újraindítását igényli a **radiusctl** paranccsal.

RADIUS leállítása és újraindítása

Ha a RADIUS szerver /etc/radius/radiusd.conf konfigurációs fájlja, vagy az /etc/radius/authorization/default.policy és /etc/radius/authorization/default.auth alapértelmezett felhatalmazási fájlok egyike módosításra kerül, akkor újra kell indítani a **radiusd** démonokat. Ez kezelhető az SMIT-ből és a parancssorból.

A RADIUS szerver elindításához, újraindításához és leállításához használja a következő parancsokat:

```
radiusctl start
radiusctl restart
radiusctl stop
```

A RADIUS újraindítása azért szükséges, mert démonnak újra kell építenie a memóriatáblát a fenti konfigurációs fájlokban található alapértelmezett attribútumok alapján. A helyi felhasználókhoz osztott memória kerül alkalmazásra és a helyi felhasználók táblázatának összeállítása teljesítményokok miatt csak a démon inicializálásakor történik meg.

On-demand szolgáltatás:

Szükség szerint több RADIUS hitelesítési és számlázási szerver démont is elindíthat.

Minden démon más porton hallgatózik. A radiusd.conf alapbeállításban az 1812-es portot jelöli ki a hitelesítéshez, és az 1813-as portot a számlázáshoz. Ezek az IANA által kiosztott portszámok. A radiusd.conf módosításával azonban ezek a portszám megváltoztathatók, és szükség szerint további portszámok is kijelölhetők. Győződjön meg róla, hogy a hozzárendelt portszámokat nem használja más szolgáltatás. Ha a radiusd.conf fájl **Authentication_Ports** és **Accounting_Ports** mezőiben több portszám meg van adva, akkor minden porthoz külön **radiusd** démon indul el. Az egyes démonok a hozzájuk rendelt porton hallgatnak.

RADIUS konfigurációs fájlok

A RADIUS démon számos konfigurációs fájlt használ. A RADIUS csomagban megtalálható ezen fájlok mintaváltozata.

Minden konfigurációs fájl a root felhasználó és a security csoport tulajdona. A szótárfájl kivételével mindegyik konfigurációs fájl szerkeszthető a SMIT segítségével. A szervert a változtatások életbe léptetéséhez újra kell indítani.

radiusd.conf fájl:

A radiusd.conf fájl a RADIUS konfigurációs paramétereit tartalmazza.

A RADIUS alapértelmezésben az /etc/radius könyvtárban keresi a radiusd.conf fájlt. A konfigurációs fájl bejegyzéseit a fájlban látható formátumban kell megadni. A RADIUS csak az érvényes kulcsszavakat és értékeket fogadja el; ha ezek közül akár az egyik is érvénytelen, akkor az alapbeállítást használja. A RADIUS démon indításakor ellenőrizze, hogy a SYSLOG kimenete tartalmaz-e a konfigurációs paraméterek hibájára utaló bejegyzéseket. Nem minden hiba vezet ugyanis a szerver leállásához.

Ezt a fájlt olvasás- és írásvédetté kell tenni, mert befolyásolja a hitelesítési és számlázási szerverek viselkedését, és titkos adatokat is tartalmazhat.

Fontos: A radiusd.conf fájl szerkesztésekor ne módosítsa a bejegyzések sorrendjét. A SMIT képernyői az értékek fájlbeli sorrendre támaszkodnak.

A következő rész a radiusd.conf fájlra mutat be egy példát:

```
#------------------------------------------------------------------#
#              CONFIGURATION FILE                                  #
#                                                                  #
#  By default RADIUS will search for radiusd.conf in the           #
#  /etc/radius directory.                                          #
#                                                                  #
#  Configuration file entries need to be in the below              #
#  formats.  RADIUS will accept only valid "Keyword : value(s)",   #
#  and will use defaults, if "Keyword : value(s)" are not          #
#  present or are in error.                                        #
#                                                                  #
#  It is important to check the syslog output when launching       #
#  the radius daemons to check for configuration parameter         #
#  errors. Once again, not all configuration errors will lead to   #
#  the server stopping.                                            #
#                                                                  #
#  Lastly, this file should be appropriately read/write protected, #
#  because it will affect the behavior of authentication and       #
#  accounting, and confidential or secretive material may          #
#  exist in this file.                                             #
#                                                                  #
#  IF YOU ARE EDITING THIS FILE, DO NOT CHANGE THE ORDER OF THE    #
#  ENTRIES IN THIS FILE. SMIT PANELS DEPEND ON THE ORDER.          #
#                                                                  #
#                                                                  #
#------------------------------------------------------------------#

#------------------------------------------------------------------#
#               Global Configuration                               #
#                                                                  #
#   RADIUSdirectory  :  This is the base directory for the RADIUS  #
#                       daemon. The daemon will search this        #
#                       directory for further configuration files. #
#                                                                  #
#   Database_location : This is the value of where the             #
#                       authentication (user ids & passwords)      #
#                       will be stored and retrieved.              #
#                       Valid values: Local, LDAP, UNIX            #
#                       UNIX  - User defined in AIX system         #
#                       Local - Local AVL Database using raddbm    #
#                       LDAP  - Central Database                   #
#                                                                  #
#   Local_Database   :  This indicates the name of the local       #
#                       database file to be used.                  #
#                       This field must be completed if the        #
#                       Database location is Local.                #
#                                                                  #
#   Debug_Level      :  This pair sets the debug level at which    #
#                       the RADIUS server will run.  Appropriate   #
#                       values are 0,3 or 9. The default is 3.     #
#                       Output is directed to location specified   #
#                       by *.debug stanza in /etc/syslog.conf      #
#                                                                  #
#                       Each level increases the amount of messages#
#                       sent to syslog. For example "9" includes   #
#                       the new messages provided by "9" as well   #
#                       as all messages generated by level 0 and 3.#
#                                                                  #
#                       0 : provides the minimal output to the     #
#                           syslogd log.  It sends start up        #
#                           and end messages for each RADIUS       #
#                           process.  It also logs error           #
#                           conditions.                            #
#                                                                  #
#                       3 : includes general ACCESS ACCEPT, REJECT #
#                           and DISCARD messages for each packet.  #
#                           This level provides a general audit    #
```

```
#                           trail for authentication.              #
#                                                                  #
#                       9 : Maximum amount of log data. Specific   #
#                           values of attributes while a           #
#                           transaction is passing thru            #
#                           processing and more.                   #
#                           [NOT advised under normal operations]  #
#                                                                  #
#------------------------------------------------------------------#
RADIUSdirectory   : /etc/radius
Database_location : UNIX
Local_Database    : dbdata.bin
Debug_Level       : 3
#------------------------------------------------------------------#
#                 Accounting Configuration                         #
#                                                                  #
#   Local_Accounting :  When this flag is set to ON or TRUE a file #
#                       will contain a record of ACCOUNTING START  #
#                       and STOP packets received from the Network #
#                       Access Server(NAS).  The default log file  #
#                       is:                                        #
#                           /var/radius/data/accounting            #
#                                                                  #
#  Local_accounting_loc  : /var/radius/data/accounting             #
#                          path and file name of the local         #
#                          accounting data file. Used only if Local_ #
#                          Accounting=ON. If the default is        #
#                          changed, then the path and file need to #
#                          to be created (with proper permissions) #
#                          by the admin.                           #
#                                                                  #
#------------------------------------------------------------------#
Local_Accounting  : ON
Local_Accounting_loc : /var/radius/data/accounting
#------------------------------------------------------------------#
#     Reply Message Attributes                                     #
#                                                                  #
#     Accept_Reply-Message : Sent when the RADIUS server           #
#                            replies with an Access-Accept packet  #
#                                                                  #
#     Reject_Reply-Message : Sent when the RADIUS server           #
#                            replies with an Access-Reject packet  #
#                                                                  #
#     Challenge_Reply-Message   : Sent when the RADIUS server      #
#                                 replies with an Access-Challenge #
#                                 packet                           #
#------------------------------------------------------------------#
Accept_Reply-Message :
Reject_Reply-Message :
Challenge_Reply-Message :
Password_Expired_Reply-Message :
#------------------------------------------------------------------#
#    Support Renewal of Expired Password                           #
#                                                                  #
#    Allow_Password_Renewal: YES or NO                             #
#                            Setting this attribute to YES allows  #
#                            users to update their expired password#
#                            via the RADIUS protocol. This requires#
#                            the hardware support of               #
#                            Access-Password-Request packets.      #
#------------------------------------------------------------------#
Allow_Password_Renewal  : NO
#------------------------------------------------------------------#
#    Require Message Authenticator in Access-Request               #
#                                                                  #
#    Require_Message_Authenticator: YES or NO      #
#                                   Setting this attribute to YES  #
```

```
#         checks message authenticator   #
#                                  in Access-Request packet.If not#
#        present, it will discard the   #
#                                  packet.                         #
#------------------------------------------------------------------#
Require_Message_Authenticator : NO
#------------------------------------------------------------------#
#       Servers ( Authentication and Accounting )                  #
#                                                                  #
#   Authentication_Ports : This field indicates on which port(s)   #
#                          the authentication server(s) will listen#
#                          on.  If the field is blank an           #
#                          authentication daemon will not be       #
#                          started.                                #
#                          The value field may contain more than   #
#                          one value.  Each value is REQUIRED to   #
#                          be separated by a comma ','.            #
#                                                                  #
#                          The value field must contain a numeric  #
#                          value, like "6666".  In this case a     #
#                          server daemon will listen on "6666".    #
#                                                                  #
#   Accounting_Ports     : The same as authentication_Ports.  See  #
#                          above definitions.                      #
#                                                                  #
# [NOTE] There is no check for port conflicts.  If a server is     #
#        currently running on the specified port the deamon will   #
#        error and not run.  Be sure to check the syslog output    #
#        insure that all servers have started without incident.    #
#                                                                  #
#                                                                  #
# [Example]                                                        #
#   Authentication_Ports  : 1812,6666 (No Space between commas)    #
#                                                                  #
#   In the above example a sever will be start for each port       #
#   specified.  In the case                                        #
#                                                                  #
#           6666 : port 6666                                       #
#                                                                  #
#------------------------------------------------------------------#
Authentication_Ports  : 1812
Accounting_Ports      : 1813
#------------------------------------------------------------------#
#       LDAP Directory User Information                            #
#                                                                  #
#   Required if RADIUS is to connect to a LDAP Version 3 Directory #
#   and the Database_location field=LDAP                           #
#                                                                  #
#   LDAP_User     : User ID which has admin permission to connect  #
#                   to the remote (LDAP) database. This is the     #
#                   the LDAP administrator's DN.                   #
#                                                                  #
#   LDAP_User_Pwd : Password associated with the above User Id     #
#                   which is required to authenticate to the LDAP  #
#                   directory.                                     #
#                                                                  #
#------------------------------------------------------------------#
LDAP_User        : cn=root
LDAP_User_Pwd    :
#------------------------------------------------------------------#
#       LDAP Directory Information                                 #
#                                                                  #
#   If the Database_location field is set to "LDAP" then the       #
#   following fields need to be completed.                         #
#                                                                  #
#   LDAP_Server_name     : This field specifies the fully qualified#
#                          host name where the LDAP Version 3      #
```

```
#                          Server is located.                        #
#   LDAP_Server_Port     : The TCP port number for the LDAP server #
#                          The standard LDAP port is 389.          #
#   LDP_Base_DN          : The distinguished name for search start #
#   LDAP_Timeout         : # seconds to wait for a response from   #
#                          the LDAP server                         #
#   LDAP_Hoplimit        : maximum number of referrals to follow   #
#                          in a sequence                           #
#   LDAP_Sizelimit       : size limit (in entries) for search      #
#   LDAP_Debug_level     : 0=OFF 1=Trace ON                        #
#                                                                  #
#------------------------------------------------------------------#
LDAP_Server_name          :
LDAP_Server_port          :  389
LDAP_Base_DN              :  cn=aixradius
LDAP_Timeout              :  10
LDAP_Hoplimit             :  0
LDAP_Sizelimit            :  0
LDAP_Debug_level          :  0
#------------------------------------------------------------------#
#       PROXY RADIUS Information                                   #
#                                                                  #
#                                                                  #
#   Proxy_Allow              :  ON or OFF. If ON, then the server              #
#                               can proxy packets to realms it    #
#                               knows of and the following        #
#                               fields must also be configured.   #
#   Proxy_Use_Table          :  ON or OFF. If ON, then the server              #
#                               can use table for faster          #
#                               processing of duplicate requests  #
#                               Can be used without proxy ON, but #
#                               it is required to be ON if        #
#                               Proxy_Use_Table is set to ON.     #
#   Proxy_Realm_name         :  This field specifies the realm    #
#                               this server services.             #
#   Proxy_Prefix_delim       :  A list of separators for parsing  #
#                               realm names added as a prefix to  #
#                               the username.  This list must be  #
#                               mutually exclusive to the Suffix  #
#                               delimiters.                       #
#   Proxy_Suffix_delim       :  A list of separators for parsing  #
#                               realm names added as a suffix to  #
#                               the username.  This list must be  #
#                               mutually exclusive to the Prefix  #
#                               delimiters.                       #
#   Proxy_Remove_Hops        :  YES or NO.  If YES then the       #
#                               will remove its realm name, the   #
#                               realm names of any previous hops  #
#                               and the realm name of the next    #
#                               server the packet will proxy to.  #
#                                                                 #
#   Proxy_Retry_count        :  The number of times to attempt    #
#                               to send the request packet.       #
#                                                                 #
#   Proxy_Time_Out           :  The number of seconds to wait     #
#                               in between send attempts.         #
#                                                                 #
#-----------------------------------------------------------------#
Proxy_Allow               :   OFF
Proxy_Use_Table           :   OFF
Proxy_Realm_name          :
Proxy_Prefix_delim        :   $/
Proxy_Suffix_delim        :   @.
Proxy_Remove_Hops         :   NO
Proxy_Retry_count         :   2
Proxy_Time_Out            :   30
#-----------------------------------------------------------------#
```

```
#    Local Operating System Authentication Configuration          #
#                                                                 #
#  UNIX_Check_Login_Restrictions  : ON or OFF. If ON, during      #
#                                   local operating system authen- #
#                                   tication, a call to           #
#                                   loginrestrictions() will be   #
#                                   made to verify the user has   #
#                                   no local login restrictions.  #
#                                                                 #
#-----------------------------------------------------------------#
UNIX_Check_Login_Restrictions : OFF
#-----------------------------------------------------------------#
#    Global IP Pooling Flag                                       #
#                                                                 #
#  Enable_IP_Pool : ON or OFF. If ON, then RADIUS Server will do  #
#                   IP address assignment from a pool of addresses #
#                   defined to the RADIUS server.                 #
#                                                                 #
#-----------------------------------------------------------------#
Enable_IP_Pool            :   OFF
#-----------------------------------------------------------------#
# Send Accept MA: ON or OFF. Some NAS's dislike it if Message     #
#                 Authenticators (MA's) are present in an ACCEPT  #
#                 message. Use this option to disable sending MA  #
#                 when sending an ACCEPT.                         #
#                                                                 #
# NOTE: Sometimes these same NAS's do not like custom ACCEPT      #
# messages either.                                                #
#                                                                 #
#-----------------------------------------------------------------#
Send_Accept_MA : ON
#-----------------------------------------------------------------#
#                                                                 #
# Maximum_Threads : The number of threads that will get           #
#                   spawned to handle authentication              #
#                   requests. If nothing is specified             #
#                   RADIUS defaults to 10.                        #
#                                                                 #
#-----------------------------------------------------------------#
Maximum_Threads : 99
#-----------------------------------------------------------------#
#                                                                 #
# EAP_Conversation_Timeout : The number of seconds to wait        #
#                            before a conversation becomes        #
#                            stale and gets deleted.              #
#                                                                 #
# NOTE: This prevents Denial-of-Service (DoS) attacks on the      #
#       RADIUS Authentication Server. You may need to increase    #
#       the value of this timeout if your network has high        #
#       latency.                                                  #
#                                                                 #
#-----------------------------------------------------------------#
EAP_Conversation_Timeout : 30
#-----------------------------------------------------------------#
# Global EAP-TLS (eap-tls) Configuration Settings:                #
#                                                                 #
# Examples:                                                       #
#                                                                 #
# Enable_EAP-TLS : ON or OFF. If ON, then the server              #
#                  can use OpenSSL to authenticate users          #
#                  using EAP-TLS. These users must first          #
#                  have an EAP authentication type of 13          #
#                  (or EAP-TLS). This setting is found in         #
#                  smitty, using: 'smitty rad_conf_users'         #
#                                                                 #
# NOTE: The following attributes below are completely ignored     #
#       if the above 'Enable_EAP' attribute is not 'ON'.          #
```

```
#                                                                    #
# OpenSSL_Library : /opt/freeware/lib/libssl.a(libssl.so.0.9.7)    #
# OpenSSL_Ciphers : ALL:!ADH:RC4+RSA:+SSLv2:@STRENGTH              #
# RootCA_Dir          : /etc/radius/tls                            #
# RootCA_File         : /etc/radius/tls/cacert.pem                 #
# Server_Cert_File    : /etc/radius/tls/cert-srv.pem               #
# Server_PrivKey_File : /etc/radius/tls/cert-srv.pem               #
# Server_CRL_File     : /etc/radius/tls/crl.pem                    #
#                                                                    #
# NOTE: Server_Cert_File and Server_PrivKey_File can be the        #
#       same file if the file is of the following format (but      #
#       in any order):                                             #
#                                                                    #
#       -----BEGIN RSA PRIVATE KEY-----                            #
#       Proc-Type: 4,ENCRYPTED                                     #
#       <rsa private key data here>                                #
#       -----END RSA PRIVATE KEY-----                              #
#       -----BEGIN CERTIFICATE-----                                #
#       <certificate data here>                                    #
#       -----END CERTIFICATE-----                                  #
#                                                                    #
#--------------------------------------------------------------------#
Enable_EAP-TLS      : ON
OpenSSL_Library     : /opt/freeware/lib/libssl.a(libssl.so.0.9.7)
OpenSSL_Ciphers     : ALL:!ADH:RC4+RSA:+SSLv2:@STRENGTH
RootCA_Dir          : /etc/radius/tls
RootCA_File         : /etc/radius/tls/radiusdcacert.pem
Server_Cert_File    : /etc/radius/tls/cert-srv.pem
Server_PrivKey_File : /etc/radius/tls/cert-srv.pem
Server_CRL_File     :
```

Az egyes felhasználók EAP hitelesítési módszere az SMIT-vel állítható be. Az EAP módszerek beállításához az egyes felhasználókhoz végezze el az alábbi lépéseket:

```
Radius szerver
     -> Felhasználók beállítása
          -> Helyi adatbázis
             LDAP címtár
                  -> Felhasználó hozzáadása
                     Felhasználó jellemzőinek módosítása/megjelenítése
                          ->
                          Login User ID [ ]
                          EAP Type [0 2 4]
                          Password Max Age
```

Az EAP típus kijelölése után az alábbiak közül választhat:

0 Nincs

2 MD5 - felhívás

4 TLS

A kijelölt EAP módszer a radiusd.conf fájlban beállított hitelesítési módszerrel kerül összehasonlításra a hitelesítéskor.

/etc/radius/clients fájl:

A clients fájl azon kliensek listáját tartalmazza, amelyek jogosultak kommunikálni a RADIUS szerverrel.

Általában minden kliens, NAS vagy AP esetében az IP címet és a RADIUS szerverrel megosztott titkot kell megadni, illetve esetleg az IP-tár *tárolónevét*.

A fájl az alábbi formátumú bejegyzéseket tartalmazza:

```
<Kliens IP-címe>    <Osztott titok> <Tároló neve>
```

Egy példa bejegyzés:

```
10.10.10.1      mysecret1      floor6
10.10.10.2      mysecret2      floor5
```

Az osztott titok egy karaktersorozat, amely mind a kliensharddveren, mind a RADIUS szerveren be van állítva. Az osztott titok maximális hossza 256 byte, a kis- és nagybetűk különböznek. Az osztott titok nem kerül elküldésre egyetlen RADIUS csomagban sem, és nem soha nem küldődik át a hálózaton. A rendszeradminisztrátoroknak meg kell győződniük arról, hogy mind a két oldalon (a kliensen és a RADIUS szerveren is) pontosan került beállításra a titok. Az osztott titok a felhasználói jelszóinformációk titkosítására szolgál, és egy Üzenethitelesítési attribútum segítségével felhasználható az üzenetek integritásának ellenőrzésére.

Minden kliens osztott titkának egyedinek kell lennie az /etc/radius/clients fájlban, és mint minden jó jelszó esetében, a legjobb, ha kis- és nagybetűk, számok és szimbólumok keverékét tartalmazza. Hogy biztonságos maradhasson, érdemes legalább 16 karakter hosszúságot megadni. A /etc/radius/clients fájl nem módosítható a SMIT használatával. Az osztott titkot gyakran érdemes cserélni a szótáras támadások kivédésére.

A *poolname* annak a tárolónak a neve, amelyből a dinamikus fordítások alkalmával a globális IP-címek kiosztásra kerülnek. A *poolname* értékét a rendszeradminisztrátor adja meg a RADIUS szerver telepítése során. A *poolname* egy SMIT panelben adható hozzá a rendszerhez, a **Proxy szabályok beállítása > IP tároló > IP tároló létrehozása** menüpontból. Ez a szerveroldali IP tárkezelés alkalmával használatos.

/etc/radius/dictionary fájl:

A dictionary fájl azoknak az attribútumoknak a leírását tartalmazza, amelyeket RADIUS protokoll határoz meg, és AIX RADIUS kiszolgáló támogat.

A RADIUS démon a csomagadatok ellenőrzéséhez és létrehozásához használja. A szállító saját attribútumait szintén itt lehet megadni. A dictionary fájl tetszőleges szövegszerkesztővel módosítható. SMIT felület nem áll rendelkezésre.

Az alábbi részlet a példa dictionary fájlból származik:

```
#########################################################################
#                                                                       #
#   This file contains dictionary translations for parsing              #
#   requests and generating responses.  All transactions are            #
#   composed of Attribute/Value Pairs.  The value of each attribute     #
#   is specified as one of 4 data types.  Valid data types are:         #
#                                                                       #
#   string - 0-253 octets                                               #
#   ipaddr - 4 octets in network byte order                             #
#   integer - 32 bit value in big endian order (high byte first)        #
#   date - 32 bit value in big endian order - seconds since             #
#                                   00:00:00 GMT,  Jan.  1,  1970       #
#                                                                       #
#   Enumerated values are stored in the user file with dictionary       #
#   VALUE translations for easy administration.                         #
#                                                                       #
#   Example:                                                            #
#                                                                       #
#   ATTRIBUTE         VALUE                                             #
#   ---------------   -----                                             #
#   Framed-Protocol = PPP                                               #
#   7               = 1     (integer encoding)                          #
#                                                                       #
#########################################################################
ATTRIBUTE       User-Name                       1       string
ATTRIBUTE       User-Password                   2       string
ATTRIBUTE       CHAP-Password                   3       string
ATTRIBUTE       NAS-IP-Address                  4       ipaddr
ATTRIBUTE       NAS-Port                        5       integer
ATTRIBUTE       Service-Type                    6       integer
ATTRIBUTE       Framed-Protocol                 7       integer
```

```
ATTRIBUTE       Framed-IP-Address               8       ipaddr
ATTRIBUTE       Framed-IP-Netmask               9       ipaddr
ATTRIBUTE       Framed-Routing                  10      integer
ATTRIBUTE       Filter-Id                       11      string
.
.
.
```

Megjegyzés: A default.policy vagy a default.auth fájlban (illetve egy adott felhasználói_azonosító.policy vagy felhasználói_azonosító.auth fájlban) megadott attribútumoknak a helyi AIX dictionary konfigurációs fájlnak megfelelő, érvényes RADIUS attribútumoknak kell lenniük. Ha egy attribútum nem található meg a szótárban, akkor a **radiusd** démon nem töltődik be, és a hibaüzenet bekerül a rendszernaplóba.

Megjegyzés: Ha módosítja a dictionary, a default.policy vagy a default.auth fájlt, akkor a változások életbe léptetéséhez újra kell indítani a RADIUS démonokat a **stopsrc** és **startsrc** parancsokkal, vagy a SMIT eszközzel.

/etc/radius/proxy fájl:

Az /etc/radius/proxy a proxy szolgáltatáshoz tartozó konfigurációs fájl. Ez a fájl képezi le az ismert tartományokat, amelyekbe a proxy szerver csomagokat továbbíthat.

Az /etc/radius/proxy fájl a tartományba érkező csomagokat kezelő szerver IP címét és a két szerver közötti osztott titok nevét rendeli a tartománynevekhez.

A fájl az alábbi, SMIT-vel módosítható mezőket tartalmazza:

- **Tartomány neve**
- **Következő állomás IP címe**
- **Osztott titok**

Példa egy /etc/radius/proxy fájlra:

Megjegyzés:

Az osztott titoknak érdemes legalább 16 karakter hosszúnak lennie. A következő RADIUS állomáson be kell állítani ugyanezt az osztott titkot.

```
# @(#)91 1.3  src/rad/usr/sbin/config_files/proxy, radconfig, radius530 1/23/04 13:11:14
#####################################################################
#                                                                   #
#       This file contains a list of proxy realms which are         #
#       authorized to send/receive proxy requests/responses to/from #
#       this RADIUS server and their Shared secret used in encryption.#
#                                                                   #
#       The first field is the name of the realm of the remote RADIUS #
#       Server.                                                     #
#                                                                   #
#       The second field is a valid IP address for the remote RADIUS  #
#       Server.                                                     #
#                                                                   #
#       The third column is the shared secret associated with this  #
#       realm.                                                      #
#                                                                   #
#       NOTE: This file contains sensative security information and #
#             precautions should be taken to secure access to this  #
#             file.                                                 #
#                                                                   #
#####################################################################
# REALM NAME              REALM IP              SHARED SECRET
#---------------       -------------------     ----------------------
#  myRealm                10.10.10.10             sharedsec
```

Hitelesítés

A hagyományos hitelesítés egy név és egy rögzített jelszó segítségével történik, általában a felhasználó bejelentkezésekor illetve a szolgáltatás igénylésekor. A RADIUS egy hitelesítési adatbázisra támaszkodik, amely felhasználói azonosítókat, jelszavakat és más információkat tartalmaz.

A felhasználó hitelesítése történhet helyi adatbázist, UNIX jelszavak vagy LDAP használatával. Az adatbázis helyét a szerveren található /etc/radius/radiusd.conf fájl beállításai határozzák meg. A telepítéskor megadott értéket a SMIT eszközzel módosíthatja. A RADIUS konfigurációs fájljairól további információk: “RADIUS konfigurációs fájlok” oldalszám: 309

Felhasználói adatbázisok:

A RADIUS szoftver többféle adatbázisban képes tárolni a felhasználói információkat.

A felhasználói információk tárolására használhat helyi, UNIX vagy LDAP adatbázist.

UNIX:

HA a UNIX hitelesítés lehetőséget választja, akkor a RADIUS szerver a felhasználók hitelesítéséhez a helyi rendszer hitelesítési eljárását használja fel.

Helyi UNIX hitelesítés alkalmazásához szerkessze a radiusd.conf fájl **database_location** mezejét, vagy az SMIT Adatbázis helye mezejében válassza ki a UNIX értéket. Ez a hitelesítési módszer meghívja az UNIX **authenticate()** alkalmazás programozási felületet (API) felhasználói azonosító és jelszó hitelesítéséhez. A jelszavak a UNIX által is használt adatfájlban, az /etc/passwords fájlban tárolódnak. A felhasználói azonosítók és jelszavak az **mkuser** paranccsal vagy a SMIT eszközzel hozhatók létre.

UNIX adatbázis használatához az **Adatbázis helye** mezőben válassza ki a UNIX értéket, az alább látható módon:

```
    Configure Server

RADIUS Directory            /etc/radius
*Database Location           [UNIX]
Local AVL Database File Name [dbdata.bin]
Local Accounting             [ON]

Debug Level                  [3]
.
.
.
```

Helyi:

Ha a radiusd.conf fájl **database_location** mezeje vagy a SMIT Adatbázis helye bejegyzése tartalmazza a Local szót, akkor a RADIUS szerver az összes felhasználói azonosítót és jelszót az /etc/radius/dbdata.bin tárolja.

A helyi felhasználó adatbázis egy egyszerű szöveges fájl, amely a felhasználói azonosítókat és jelszavakat tartalmazza. A jelszavak kivonatolt formában kerülnek mentésre. A kivonatolás címzési eljárás, amellyel közvetlenül elérhetőek a memóriában tárolt adatok. A felhasználói jelszavak hozzáadásához, törléséhez vagy módosításához futtassa a **raddbm** parancsot, vagy használja a SMIT eszközt. A **radiusd** démon indításkor beolvassa a radiusd.conf fájlt, és betölti a memóriába a felhasználói azonosítókat és jelszavakat.

Megjegyzés: A felhasználói azonosítók maximális hossza 253 karakter, a jelszavak legfeljebb 128 karakter hosszúak lehetnek.

A helyi felhasználói adatbázis használatához az **Adatbázis helye** mezőben válassza ki a Helyi értéket, az alább látható módon:

```
    Configure Server

RADIUS Directory            /etc/radius
*Database Location           [Local]
Local AVL Database File Name [dbdata.bin]
Local Accounting             [ON]

Debug Level                  [3]
.
.
.
```

LDAP:

A RADIUS az LDAP 3-as változatát használhatja a távoli felhasználói adatok tárolásához.

A RADIUS a felhasználói adatok távoli eléréséhez az 3-as változatú LDAP API-kat használja. A RADIUS akkor használja az LDAP 3-as változatot, ha az /etc/radiusd.conf fájl **database_location** mezejének értéke LDAP, és a szervernév, az LDAP adminisztrátor felhasználói azonosító és az LDAP adminisztrátor jelszó be van állítva.

AIX az IBM Tivoli Directory Server által biztosított és támogatott LDAP 3-as változat kliens könyvtárait használja. Az LDAP egy jól skálázható protokoll, melynek használatát a felhasználói- és folyamatadatok központi tárolása és a RADIUS könnyű adminisztrációja indokolja. A RADIUS adatok megtekintése az **ldapsearch** parancssori segédprogrammal lehetséges.

Ahhoz, hogy a RADIUS képes legyen használni, az LDAP szervert be kell állítani és megfelelően kell adminisztrálni.

A RADIUS szerver tartalmazza a RADIUS séma, objektumok és attribútumok felvételéhez szükséges LDAP ldif fájlokat, de az LDAP telepítése és konfigurálása az adminisztrátor feladata.

A RADIUS LDAP objektumok számára egy külön utótag jön létre a RADIUS LDAP objektumok használatához. Ez az utótag a cn=aixradius nevű tároló, amely két objektumosztályt tartalmaz. Ennek részletes leírása az alábbi részben található: “RADIUS LDAP szerverkonfiguráció” oldalszám: 320. Alkalmazza a RADIUS által biztosított ldif fájlt az utótag és a RADIUS séma létrehozásához.

Ha hitelesítési adatbázisnak az LDAP-t választja, akkor az alábbi szolgáltatásokat kapja:

1. A felhasználói adatbázis az összes RADIUS szerverről látszik és elérhető
2. Az aktív felhasználók listája lekérdezhető
3. Korlátozható az egy felhasználói azonosítóval engedélyezett bejelentkezések száma
4. Felhasználónként konfigurálható **EAP** típus
5. A jelszó lejárati dátuma

Az LDAP adatbázis használatához az **Adatbázis helye** mezőben válassza ki az LDAP értéket, az alább látható módon:

```
    Configure Server

RADIUS Directory            /etc/radius
*Database Location           [LDAP]
Local AVL Database File Name [dbdata.bin]
Local Accounting             [ON]

Debug Level                  [3]
.
.
.
```

Kapcsolódó tájékoztatás:

IBM Directory Server

RADIUS LDAP szerverkonfiguráció:

Az LDAP felhasználó hitelesítés beállításához módosítani kell az LDAP szerver sémát. Az LDAP rendszeradminisztrátornak fel kell vennie az AIX RADIUS által meghatározott attribútumokat és objektumosztályokat az LDAP címtárba az LDAP RADIUS felhasználók megadása előtt.

Egy utótagot is létre kell hozni az LDAP címtárban. A RADIUS utótag neve cn=aixradius. Az utótag egy olyan megkülönböztetett név, amely a címtár hierarchia egyik csúcsbejegyzését jelöli.

Az utótag hozzáadásakor az LDAP címtárban létrejön egy üres tároló. A *tároló* egy üres bejegyzés, amely a névtér felosztásához használható. A tárolók annyiban hasonlatosak a fájlrendszer könyvtáraihoz, hogy további címtárbejegyzések tartozhatnak alájuk. A felhasználói profil információk ezután a SMIT eszközzel adhatók az LDAP címtárhoz. Az LDAP adminisztrátori azonosító és jelszó az /etc/radius/radiusd.conf fájlban tárolódik, és szintén a SMIT eszközzel konfigurálható a RADIUS szerveren.

Az LDAP címtárbejegyzésekben tárolt információk a sémán belül objektumosztályok segítségével rendszerezhetők. Egy objektumosztály kötelező és elhagyható attribútumok határoznak meg. Az attribútumok típus=érték alakú párok, ahol a típust egy egyedi objektumazonosító (OID) határozza meg, az érték pedig kötött szintaxissal rendelkezik. Az LDAP címtár minden egyes bejegyzése az objektum egy példánya.

Megjegyzés: Maga az objektumosztály nem határoz meg címtárinformációs fát vagy névteret. Ez csak a bejegyzések létrehozásával és a megkülönböztetett nevek objektumosztályok-példányokhoz rendelésével történik meg. Például ha egy tároló objektumosztályhoz hozzárendel egy egyedi megkülönböztetett nevet, akkor az két további olyan bejegyzéshez is hozzárendelhető, amelyek a szervezeti egység objektumosztály példányai. Az eredmény egy fa-jellegű struktúra vagy névtér.

A RADIUS szerver objektumosztályait egy ldif fájl tartalmazza. Bizonyos attribútumok meglévő LDAP séma attribútumok, míg mások a RADIUS szerverre egyedileg jellemzőek. Az új RADIUS objektumosztályok strukturálisak és absztraktak.

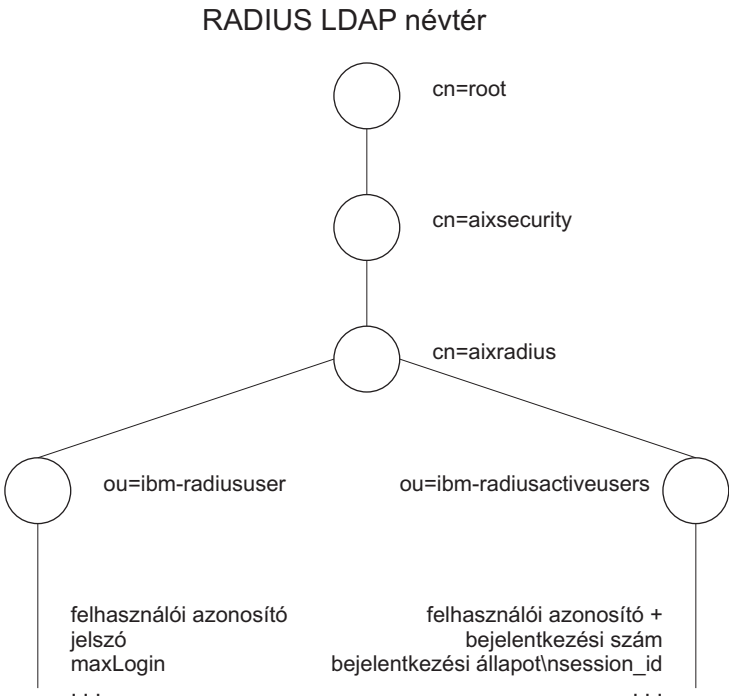
A RADIUS az LDAP szerverhez kapcsolódáshoz illetve az LDAP adminisztrátori DN és az jelszó megadásához biztonsági okokból a RAM-MD5 hitelesítés módszert alkalmazó **ldap_bind_s** SASLP API hívást használja. Ez a módszer a jelszó helyett csak egy kivonatot küld át a hálózaton. A CRAM-MD5 biztonsági mechanizmus használatához sem a kliensen, sem a szerveren nincs szükség különleges beállítások elvégzésére.

Megjegyzés: Az objektumosztályokban található összes attribútum egyértékű.

RADIUS LDAP névtér:

A RADIUS LDAP névtér a cn=aixradius tárolót tartalmazza a hierarchiája legtetején. A cn=aixradius alatt két szervezeti egység (OU-k) helyezkedik el. A szervezeti egységek olyan tárolók, melyek elősegítik a bejegyzések egyedivé tételét.

Az alábbi ábra a RADIUS LDAP sémát szemlélteti. Az ábra a tárolókat és szervezeti egységeket körökkel, az ágakat vonalakkal jelöli. Az aixradius tároló középen két szervezeti egységbe ágazik el: ibm-radiususer és ibm-radiusactiveusers. Az ibm-radiususer tároló alá tartozik implicit módon a userid, password és maxLogin tároló. Az ibmradiusactiveusers tároló alá tartoznak implicit módon a userid +, login number, login status és session_id tárolók. Az aixradius tároló fölött az aixsecurity tároló található, a root tároló pedig a csúcson.



16. ábra: RADIUS LDAP névtér

LDAP névtér sémafájlok:

Az LDAP sémafájlok obajktumosztályokat és RADIUS attribútumokat adnak meg az LDAP névtérnek.

Az /etc/radius/ldap könyvtárban az alábbi LDAP séma fájlok találhatók:

IBM.V3.radiusbase.schema.ldif

Ez a fájl a felsőszintű objektumosztályokat adja meg a RADIUS szerver számára (cn=aixradius). A fájl ezenkívül az alábbi ágakat hozza létre a cn=aixradius objektumosztály alatt:

```
ou=ibm-radiususer
ou=ibm-radiusactiveusers
```

A szükséges további információk megadásához futtassa az alábbi parancsot:

```
ldapadd -D ldap_adminisztrátor_azonosító -w jelszó -i /etc/radius/ldap/IBM.V3.radiusbase.schema.ldif
```

Ezt a parancsot futtathatja az LDAP szerveren, vagy futtathatja távolról a **-h** (hosztrendszer neve) kapcsolóval.

IBM.V3.radius.schema.ldif

Ez a fájl a RADIUS attribútumokat és az objektumosztályokat adja meg.

Új RADIUS attribútumok és objektumosztályok felvételéhez adja ki az alábbi parancsot:

```
ldapmodify -D ldap_adminisztrátor_azonosító -w jelszó -i /etc/radius/ldap/IBM.V3.radius.schema.ldif
```

A SMIT eszközzel állítsa be az adatbázis helyét LDAP értékre, és adja meg az LDAP szerver nevét és az adminisztrátori jelszót. Ezután hozzáadhatja a RADIUS LDAP felhasználókat a könyvtárhoz az SMIT-n keresztül.

Felhasználói profil objektumosztály:

Mielőtt a RADIUS szerver hitelesíthetne egy felhasználót a rendszernek, létre kell hozni az LDAP felhasználói profilokat. A profil egy felhasználói azonosítót és egy jelszót tartalmaz.

A felhasználói profil objektumok tartalmazzák a hálózati hozzáféréssel rendelkező egyének hitelesítési információit. Az **ibm-radiusUserInstance** objektumosztályt a démon valós időben éri el az LDAP API hívásokkal. A DN kezdetét jelentő egyedi mező a felhasználói azonosító. A **MaxLoginCount** mező azt határozza meg, hogy egy LDAP felhasználó hányszor jelentkezhet be.

Aktív bejelentkezési lista objektumosztály:

Az aktív LDAP bejelentkezések listája a jelenleg bejelentkezett felhasználókra vonatkozó adatokat képviseli.

Egy felhasználóhoz több rekord is tartozhat a login_number = 1 rekordtól kezdődően az 5-ös MaxLoginCount rekordig. A munkamenet-azonosító a start_accounting RADIUS üzenetből származik. A részben kitöltött rekordok az **ibm-radiusUserInstance** objektummal együtt jönnek létre. Ez azt jelenti, hogy a RADIUS számlázási csomagok megérkezte előtt a legtöbb mező üres. Miután a start_accounting RADIUS üzenet megérkezik, az **ibm-radiusactiveusers** objektum tartalma frissül. Ezzel jelzi, hogy a felhasználó be van jelentkezve, és az egyedi munkamenet információk rögzítésre kerültek a megfelelő bejelentkezési szám alatt. A stop_accounting üzenet fogadása után az aktív bejelentkezések listájában található rekord tartalma törlődik. Az aktív bejelentkezés rekord tartalmának változása jelzi, hogy a felhasználó kijelentkezett. A számlázást indító és befejező üzenetekben a munkamenet száma megegyezik. Az objektumosztályt az LDAP API hívások valós időben érik el.

Jelszóhitelesítési protokoll:

A Jelszóhitelesítési protokoll (**PAP**) elkészíti a jelszó kivonatát egy a kliens és a szerver által egyaránt előállítható értéken alapuló MD5 algoritmussal.

Az alábbi módon működik:

1. A felhasználó jelszavát tartalmazó csomagokban az Authenticator mező egy 16 byte-os véletlenszámot, az úgynevezett kérés hitelesítőt tartalmazza.
2. Az Authenticator mező tartalma és a kliens osztott titka bekerül egy MD5 kivonatba. Az eredmény egy 16 byte hosszú kivonat.
3. Az eljárás a felhasználó által megadott jelszót nullákkal 16 karakter hosszúra egészíti ki.
4. Ezután a 2. lépésben előállt kivonaton és a kiegészített jelszón kizáró vagy (XOR) művelet kerül végrehajtásra. Ennek eredményét küldi el csomagban *user_password* attribútumként.
5. A RADIUS szerver kiszámítja ugyanazt a kivonatot, mint a 2. lépésben.
6. A kivonat és a 4. lépésben előállt eredmény között XOR műveletet hajt végre, és így megkapja a jelszót.

Egyeztetésre felszólításos hitelesítési protokoll:

A RADIUS támogatja a PPP **CHAP** használatát jelszóvédelemhez.

A CHAP protokoll alkalmazása során a kliens nem küldi el a felhasználó jelszavát a hálózaton keresztül. Ehelyett a jelszó MD5 kivonatát küldi el, a RADIUS szerver pedig rekonstruálja a kivonatot a felhasználói információkból a tárolt jelszót is beleértve, majd összeshasonlítja a kliens által küldött értékkel.

Kiterjeszthető hitelesítési protokoll:

A Bővíthető hitelesítési protokollt (**EAP**) több hitelesítési módszer támogatására tervezték.

Az **EAP** meghatározza a kliens és a hitelesítő szerver között kommunikáció keretét, azonban nem adja meg a hitelesítési adatok tartalmát. Ezt a tartalmat a hitelesítéshez használt **EAP** módszer írja elő. A gyakori **EAP** módszerek:

- MD5 felhívás
- Egy időre szóló jelszó
- Általános token kártya
- Szállítási réteg biztonság (TLS)

Az **EAP** adatok RADIUS szerver és kliens közötti átvitele különleges **EAP** RADIUS attribútumok segítségével történik. Ezeket az **EAP** adatokat a RADIUS szerver ezután közvetlenül elküldheti az **EAP** műveleteket megvalósító háttérszervereknek.

Az AIX RADIUS szerver csak az EAP-TLS és az MD5-felhívásos EAP metódust támogatja.

A hitelesítéshez használt EAP módszert felhasználói szinten lehet beállítani az LDAP vagy a helyi adatbázis felhasználóhoz tartozó bejegyzésében.

Az EAP alapértelmezésben minden felhasználónál ki van kapcsolva.

Felhatalmazás

A felhasználók RADIUS felhatalmazási attribútumait a default.auth és default.policy felhatalmazási házirend fájlok határozzák meg.

A felhatalmazási attribútumok az RFC-nek megfelelő érvényes RADIUS protokoll attribútumok, amelyek meghatározása a /etc/radius/dictionary fájlban található. A felhatalmazás elhagyható, használata a hardver NAS illetve elérési pont konfigurációjától függ. Ha a felhatalmazás szükséges, akkor konfigurálja a felhatalmazási attribútumokat. A felhatalmazásra csak a sikeres hitelesítést követően kerül sor.

A házirendek konfigurálható felhasználói tulajdonság-érték párok, amelyek a felhasználó hálózatelérését vezérlik. Egy házirend lehet a RADIUS szerverre nézve globális, vagy egyetlen felhasználóra jellemző.

Két felhatalmazási konfigurációs fájlt tartalmaz: /etc/radius/authorization/default.auth és default.policy. A default.policy fájl hatáskörébe a bejövő elérési kérés csomagok tartoznak. A fájl tulajdonság-érték párokat tartalmaz, melyek kezdetben üresek. Ezek konfigurálásával kell megadni a kívánt beállításokat. A hitelesítés után a házirendtől függ az, hogy a szerver hozzáférés elutasítva vagy hozzáférés engedélyezve csomagot küld a kliensnek.

Ezenkívül minden felhasználóhoz tartozhat egy *felhasználói_azonosító*.policy fájl. Ha a felhasználó rendelkezik egyedi házirend-fájllal, akkor a szerver először az ebben tárolt attribútumokat vizsgálja meg. Ha a *felhasználói_azonosító*.policy fájlban található tulajdonság-érték párok között nem talál teljesen egyezőt, akkor lép tovább a default.policy fájl ellenőrzésére. Ha a hozzáférés kérés csomaghoz tartozó attribútumok egyik fájl tartalmának sem felelnek meg, akkor a szerver hozzáférés elutasítva csomagot küld vissza. Ha talál egyezést az egyik vagy a másik fájlban, akkor egy hozzáférés engedélyezve csomagot küld a kliensnek. Ez a szerkezet hatékonyan valósítja meg a kétszintű házirendet.

A default.auth fájl tulajdonság-érték párokat tartalmaz, melyeket a szerver a házirend ellenőrzése után visszaküld a kliensnek. A default.auth fájl szintén tulajdonság-érték párokat tartalmaz, melyek kezdetben üresek. Ezek konfigurálásával kell megadni a kívánt beállításokat. Az SMIT eszközzel vagy a default.auth fájl közvetlen szerkesztésével konfigurálja a kívánt felhatalmazási attribútum-beállításokat. A szerver az értéket tartalmazó összes tulajdonságot automatikusan visszaküldi a NAS szervernek egy hozzáférés engedélyezve csomagban.

Egyedi válasz felhatalmazás attribútumokat is létrehozhat az egyes felhasználók számára. Ehhez hozzon létre egy fájlt, amelynek neve a felhasználónévből és az .auth kiterjesztésből áll, például: ***felhasználói_azonosító***.auth. Ennek az egyedi fájlnak az /etc/radius/authorization könyvtárban kell lennie. A fájlok létrehozásához a SMIT egy külön panelt biztosít.

A szerver a felhasználók felhatalmazási attribútumait a default.auth vagy global.auth fájlban található alapértelmezett felhatalmazási attribútumokkal együtt visszaküldi egy hozzáférés engedélyezve csomagban.

Ha a tulajdonságok megegyeznek a default.auth és a *felhasználói_azonosító*.auth fájlokban, akkor a felhasználó beállításai felülbírálják az alapértékeket. Ez lehetővé teszi globális felhatalmazási attribútumok (szolgáltatások vagy erőforrások) használatát, de teret biztosít a felhasználók felhatalmazásának pontosabb, egyedi meghatározásának.

Megjegyzés: A global.auth fájl segítségével egyesíthetők a felhatalmazási attribútumok a felhasználóra jellemző felhatalmazási attribútumokkal a default.auth fájl helyett, hacsak nem más egyesítési viselkedés a kívánatos.

Az AIX v6.1 6100-02 technológiai szintű változatával kezdődően a RADIUS támogatja aglobal.auth felhatalmazási fájlt. Ez a fájl helyettesíti és kiterjeszti a felhasználóra jellemző felhatalmazási attribútumok egyesítésének eredeti célját (ahogy az a user_id.auth fájlokban meg van adva) a globális felhatalmazási attribútumok halmazával.

A default.auth fájllal ellentétben - amely felülírásra került a felhasználóra jellemző felhatalmazási fájlokban található hasonló attribútumokkal - a user_id.auth fájl egyesül ezekkel az atribútumokkal, ezáltal nagyobb rugalmasságot nyújt a felhasználók felhatalmazásának karbantartásában.

Ha az attribútumok a default.auth és a user_id.auth file fájlban megegyeznek, akkor a felhasználó értékei felülírják az alapértelmezett értékeket. Az alapértelmezett értékek felülírásával néhány alapértelmezett felhatalmazási attribútum (szolgáltatás vagy erőforrás) használható az összes felhasználóhoz, de teret biztosít a felhasználók felhatalmazásának pontosabb, egyedi meghatározására.

Ugyanez igaz a global.auth fájl attribútumaira, azzal a kivétellel, hogy a user_id.auth attribútumok nem írják felül. Ehelyett a két fájl attribútumai egyesítésre kerülnek. Ez akkor hasznos, ha szállítóra jellemző attribútumokat ad meg (VSA).

A felhatalmazási folyamat a következő:

1. A démon indításakor beolvasásra kerülnek az /etc/radius/authorization/default.policy, default.auth és default.auth fájlban található alapértelmezett házirend- és jogosultsági listák a memóriába.
2. A felhasználói azonosító és jelszó hitelesítése.
3. A bejövő csomagban található tulajdonság-érték párok ellenőrzése.
   a. Az egyéni ***felhasználói_azonosító***.auth fájl ellenőrzése.
   b. Ha nincs egyezése, akkor a default.policy fájl ellenőrzése.
   c. Ha nincs egyezés, akkor hozzáférés elutasítva csomag küldése a kliensnek.
4. A felhasználó felhatalmazási attribútumainak alkalmazása (ha vannak).
   a. Az /etc/radius/authorization/***felhasználói_azonosító***.auth és default.auth fájlok beolvasása, a tartalmuk összehasonlítása.
   b. A felhasználói fájl beállításai felülbírálják az alapértelmezett beállításokat.
   c. Egyesítse az eredményül kapott attribútumokat a global.auth fájlban találhatókkal.
5. A felhatalmazási attribútumok visszaküldése egy hozzáférés engedélyezve csomagban.

Számlázás

A RADIUS számlázási szerver felelős a számlázási kérések fogadásáért, és a sikeres vételt illetve a számlázási adatok mentését igazoló nyugta visszaküldéséért.

A helyi számlázást a radiusd.conf konfigurációs fájlban lehet engedélyezni.

Ha egy kliens be van állítva a RADIUS számlázás használatára, akkor a szolgáltatás megkezdésekor létrehoz egy ACCOUNTING_START csomagot, amely tartalmazza a felhasználónak nyújtott szolgáltatás leírását, a felhasználó azonosítóját, és a kezdés időpontját. Ezután a kliens elküldi a csomagot a RADIUS számlázási szervernek, amely visszaküldi a csomag fogadását igazoló nyugtát. A szolgáltatás végeztével a kliens előállít egy ACCOUNTING_STOP csomagot, ami tartalmazza a felhasználónak nyújtott szolgáltatást, és tetszés szerint statisztikákat, mint például az eltelt időt, a küldött és fogadott byte-ok száma, vagy a bemenő és kimenő csomagok száma. Ezután a kliens elküldi a csomagot a RADIUS számlázási szervernek, amely visszaküldi a csomag fogadását igazoló nyugtát.

Az ACCOUNTING_START illetve ACCOUNTING_STOP kéréseket a kliens a hálózaton keresztül küldi el a RADIUS szervernek. A kliensnek ajánlott addig próbálkozni az ACCOUNTING_REQUEST csomag küldésével, amíg vissza nem érkezik a nyugta. Ha az elsődleges szerver leállt vagy elérhetetlen, akkor a kliens továbbíthatja a kérést egy másik szervernek illetve szervereknek is a proxy szolgáltatás segítségével. A proxyszolgáltatásokról további információkat a következő helyen talál: “Proxy szolgáltatások” oldalszám: 325.

A szerver a számlázási adatokat a helyi /etc/var/radius/data/accounting fájlban rögzíti, a szabványos *attribútum=érték* RADIUS formátumban. A bejegyzés a csomag adatait tartalmazza, egy időpecséttel kiegészítve. Ha a RADIUS számlázási szervernek nem sikerül lehegyezni a számlázási csomagot, akkor nem küldi vissza a kliensnek a **Számlázási_válasz** nyugtát, és a hibára vonatkozó információkat naplózza a syslog fájlban.

/var/radius/data/accounting fájl:

A /var/radius/data/accounting elfogja amit a kliens az ACCOUNTING START és az ACCOUNTING STOP csomagokban küld.

Az /var/radius/data/accounting fájl a telepítést követően üres. A fájl a kliensek által küldött ACCOUNTING_START és ACCOUNTING_STOP csomagok adatai tartalmazza.

Az alábbi részlet egy példa arra, hogy milyen típusú adatokat írhat az AIX RADIUS szerver a /var/radius/data/accounting fájlba. A fájl tartalma azonban nagyban függ az adott rendszer beállításaitól.

Megjegyzés:

- Biztosítsa, hogy a /var fájlrendszeren kellően nagy szabad terület álljon rendelkezésre a számlázási adatok tárolására.
- A fájl adatainak elemzésére felhasználhat harmadik féltől származó perl parancsfájlokat. A http://www.pgregg.com/projects/radiusreport webhelyen olyan példa parancsfájlok találhatók, amelyek jelentéseket készítenek a számlázási adatokból.
- A számlázási csomagok képesek együttműködni a proxy szolgáltatással.

```
Thu May 27 14:43:19 2004
 NAS-IP-Address = 10.10.10.1
 NAS-Port = 1
 NAS-Port-Type = Async
 User-Name = "rod"
 Acct-Status-Type = Start
 Acct-Authentic = RADIUS
 Service-Type = Framed-User
 Acct-Session-Id = "0000000C"
 Framed-Protocol = PPP
 Acct-Delay-Time = 0
 Timestamp = 1085686999

Thu May 27 14:45:19 2004
 NAS-IP-Address = 10.10.10.1
 NAS-Port = 1    <-- rod fizikailag csatlakozott a hardver 1-es portjához
 NAS-Port-Type = Async
 User-Name = "rod"
 Acct-Status-Type = Stop
 Acct-Authentic = RADIUS
 Service-Type = Framed-User
 Acct-Session-Id = "0000000C"   <-- A munkamenet azonosítók megegyeznek
 Framed-Protocol = PPP
 Framed-IP-Address = 10.10.10.2   <-- rod IP címe
 Acct-Terminate-Cause = User-Request  <-- A felhasználó megszakította a munkamenetet
 Acct-Input-Octets = 4016
 Acct-Output-Octets = 142
 Acct-Input-Packets = 35
 Acct-Output-Packets = 7
 Acct-Session-Time = 120  <--- másodpercekben
 Acct-Delay-Time = 0
 Timestamp = 1085687119  <--- "rod" csak 120 másodpercre (két percre)
jelentkezett be
```

Proxy szolgáltatások

A proxy szolgáltatások lehetővé teszik a RADIUS szerver számára, hogy továbbítsa a NAS szervertől származó kéréseket egy másik RADIUS szerver felé, majd visszaküldje a válaszüzenetet a NAS-nak. A proxyszolgáltatások egy tartományneven alapulnak.

A RADIUS szerver egyidejűleg proxy szerverként és háttérszerverként is képes működni. Ez a mechanizmus a számlázási és hitelesítési csomagokra egyaránt alkalmazható. A proxy szolgáltatás alapértelmezésben le van tiltva a radiusd.conf fájlban.

Tartományok:

A tartományok a csomagok User-Name attribútumának szokásos tartalma elé vagy mögé illesztett azonosítók, melyek segítségével a RADIUS szerver azonosíthatja a szervert, amellyel a hitelesítési illetve számlázási folyamat elindításához kapcsolatba kell lépni.

Az alábbi példa a tartományok használatát mutatja be RADIUS-szal:

A *Joe* felhasználó az *XYZ* társaság alkalmazottja Sacramento-ban. Ennek a területnek a tartományazonosítója *SAC*. Azonban *Joe* jelenleg egy kiküldetésen tartózkodik New York City-ben. New York City tartományazonosítója *NYC*. Amikor *Joe* betárcsáz a *NYC* tartományba, akkor a kliense a *SAC/Joe* User-Name attribútumot továbbítja. Ez tudatja a *NYC* RADIUS tartományszerverrel, hogy ezt a csomagot a *SAC* tartománybeli felhasználók hitelesítéséért és számlázásáért felelős szervernek kell továbbítani.

Tartomány user-name attribútum:

A hitelesítési vagy számlázási csomag tartományon belüli útvonala a **User-Name** attribútumtól függ. Az attribútum határozza meg azon tartományok sorrendjét, amelyeken a csomag keresztülhalad a hitelesítést illetve számlázást elvégző szerver felé.

A csomagok úgy kerülnek továbbításra, hogy a rendszer a tartományokat összefűzi a **User-Name** attribútumban. A **User-Name** attribútumba bekerülő, a csomag útját meghatározó tényleges tartományok kiválasztása a RADIUS szerkezetét megvalósító adminisztrátor feladata. A tartományállomások nevét a **User-Name** attribútum előtt és mögött is fel lehet sorolni. A tartománynevek összefűzésére leggyakrabban használt karakter a **User-Name** előtti az előtagok esetén az osztásjel (/), a **User-Name** utáni utótagok esetén pedig az és-jel (&). A szerkezetet a radiusd.conf fájlban lehet konfigurálni. A **User-Name** attribútum kiértékelése balról jobbra történik.

Az alábbi példa **User-Name** attribútum csak az előtag módszert használja:

```
USA/TEXAS/AUSTIN/joe
```

Az alábbi példa **User-Name** attribútum csak az utótag módszert használja:

```
joe@USA@TEXAS@AUSTIN
```

Az előtag és utótag módszerek együttes használata lehetséges. Azonban tartsa szem előtt, hogy a csomag által érintett tartomány állomások kiértékelése balról jobbra történik, és az összes előtag állomás kiértékelődik ez első utótag állomás előtt. A felhasználó hitelesítésének illetve a számlázási adatok rögzítésének egyetlen csomópontban kell megtörténnie.

Az alábbi példa mindkét módszer alkalmazásával ugyanazt az értéket eredményezi, mint a korábbi példák:

```
USA/joe@TEXAS@AUSTIN
```

Proxy szolgáltatás beállítása:

A RADIUS proxy konfigurációs információi a /etc/radius könyvtár proxy fájljában találhatók.

A proxy fájl a telepítés után példa bejegyzéseket tartalmaz. A proxy fájl három mezőt tartalmaz: **Tartomány neve**, **Következő állomás IP címe**, és **Osztott titok**.

A proxyszabályok beállításához válasszon az alábbiak közül:

```
    Configure Proxy Rules

Összes proxy kilistázása
Add a Proxy
Proxy jellemzőinek módosítása/megjelenítése
Proxy eltávolítása
```

Válassza az **Összes proxy kilistázása** lehetőséget a /etc/radius/proxy fájl beolvasására és a három mező oszlop formában történő megjelenítésére. Az oszlopfejlécek az alábbiak:

```
realm_name   next_hop_address  shared_secret
```

Válassza a **Proxy hozzáadása** menüpontot a következő képernyő megjelenítéséhez. Az itt megadott információk az /etc/radius/proxy fájl végéhez íródnak.

A proxylánc következő állomása a két RADIUS szerver közötti osztott titkot használja. Az osztott titkot az /etc/radius/proxy_file tartalmazza. Az osztott titoknak egyedinek kell lennie a lánc minden proxyállomásán.

Az osztott titkok létrehozásáról további információkat az “/etc/radius/clients fájl” oldalszám: 315 tartalmaz.

Egy proxy hozzáadásához válasszon a mezők közül az alábbiak szerint:

```
            Add a Proxy
*Realm Name                          []  (max 64 chars)
*Next Hop IP address (dotted decimal)    [xx.xx.xx.xx]
*Shared Secret                       []  (minimum 6, maximum 256 chars)
```

A **Proxy jellemzőinek módosítása/megjelenítése** menüpont kilistázza az elérhető tartományok neveit. A lista egy előugró képernyőn jelenik meg, és a felhasználónak ki kell választani egy tartománynevet.

A **Proxy eltávolítása** lehetőség szintén kilistázza az elérhető tartományok neveit. A lista egy előugró képernyőn jelenik meg, és a felhasználónak ki kell választani egy tartománynevet. A név kiválasztása után egy előugró képernyőn meg kell erősíteni a tartomány törlését.

Az alábbi példa a radiusd.conf fájl proxybeállításokat tartalmazó szakaszát mutatja:

```
#-----------------------------------------------------------------#
#       PROXY RADIUS Information                                  #
#                                                                 #
#                                                                 #
#   Proxy_Allow               :  ON or OFF. If ON, then the server              #
#                                can proxy packets to realms it   #
#                                knows of and the following       #
#                                fields must also be configured.  #
#   Proxy_Use_Table           :  ON or OFF. If ON, then the server              #
#                                can use table for faster         #
#                                processing of duplicate requests #
#                                Can be used without proxy ON, but #
#                                it is required to be ON if       #
#                                Proxy_Use_Table is set to ON.    #
#   Proxy_Realm_name          :  This field specifies the realm   #
#                                this server services.            #
#   Proxy_Prefix_delim        :  A list of separators for parsing #
#                                realm names added as a prefix to #
#                                the username.  This list must be #
#                                mutually exclusive to the Suffix #
#                                delimiters.                      #
#   Proxy_Suffix_delim        :  A list of separators for parsing #
#                                realm names added as a suffix to #
#                                the username.  This list must be #
#                                mutually exclusive to the Prefix #
#                                delimiters.                      #
```

```
#   Proxy_Remove_Hops         :  YES or NO.  If YES then the       #
#                                will remove its realm name, the   #
#                                realm names of any previous hops  #
#                                and the realm name of the next    #
#                                server the packet will proxy to.  #
#                                                                  #
#   Proxy_Retry_count         :  The number of times to attempt    #
#                                to send the request packet.       #
#                                                                  #
#   Proxy_Time_Out            :  The number of seconds to wait     #
#                                in between send attempts.         #
#                                                                  #
#------------------------------------------------------------------#
Proxy_Allow               :   OFF
Proxy_Use_Table           :   OFF
Proxy_Realm_name          :
Proxy_Prefix_delim        :   $/
Proxy_Suffix_delim        :   @.
Proxy_Remove_Hops         :   NO
Proxy_Retry_count         :   2
Proxy_Time_Out            :   3
```

RADIUS szerver beállítása:

A RADIUS szerver démon számos konfigurációs fájlt használ. A szerver konfigurációs információit az /etc/radius/radiusd.conf fájl tárolja. A csomagban megtalálható konfigurációs fájl az alapbeállításokat tartalmazza.

Megjegyzés: Az alábbi példán a RADIUS Szerver beállítása SMIT képernyő látható:

```
    Configure Server

RADIUS Directory            /etc/radius
*Database Location           [UNIX]
Local AVL Database File Name [dbdata.bin]
Local Accounting             [ON]
Local Accounting Directory   []

Debug Level                  [3]
Accept Reply-Message         []
Reject Reply-Message         []
Challenge Reply-Message      []
Password Expired Reply Message []
Support Renewal of Expired Passwords [NO]
Require Message Authenticator  [NO]

*Authentication Port Number  [1812]
*Accounting Port Number      [1813]

LDAP Server Name             []
LDAP Server Port Number      [389]
LDAP Server Admin Distinguished Name     []
LDAP Server Admin Password         []
LDAP Base Distinguished Name       [cn=aixradius]
LDAP Size Limit                    [0]
LDAP Hop Limit                     [0]
LDAP wait time limit               [10]
LDAP debug level                   [ 0]

Proxy Allowed                      [OFF]
Proxy Use table                    [OFF]
Proxy Realm Name                   []
Proxy Prefix Delimiters            [$/]
Proxy Suffix Delimiters            [@.]
     NOTE: prefix & suffix are mutually exclusive
Proxy Remove Hops                  [NO]
Proxy Retry Count                  [2]
Proxy Timeout                      [30]
UNIX Check Login Restrictions      [OFF]
Enable IP Pool                     [ON]
Authentication Method Sequence     [TLS, MD5]
OpenSSL Configuration File         []
```

Naplózó segédprogramok

A RADIUS szerver a SYSLOG démont használja a tevékenység- és hibainformációk naplózásához.

A naplóinformációk három szintre bonthatók:

0 Csak a problémák, hibák és a démonok indítása kerül naplózásra.

3 Az access_accept, access_reject*, discard és error üzenetek megfigyelési naplóját naplózza.

Megjegyzés: A discard üzenetek akkor kerülnek naplózásra, ha egy bejövő csomag érvénytelen, és nem jön létre válasz csomag.

9 Tartalmazza a 0-ás és 3-as naplózási szint információit, és ennél jóval többet is. A 9-es szintet csak hibakereséshez használja.

A naplózás alapértelmezett szintje a 3-as. A 3-as szint a RADIUS szerver megfigyelésének fokozására szolgál. A szerver naplózási szintjétől függően a naplóban tárolt műveletek ellenőrzésével kiszűrhetők a gyanús tevékenységre utaló nyomok. Ha egy támadó megkerüli a védelmet, akkor a SYSLOG kimenet alapján megállapítható, hogy mikor történt a betörés, és talán az elnyert jogosultság mértéke is. Ezek az információk hasznosak a jövőbeli biztonsági problémák megelőzésére tett intézkedések fejlesztéséhez.

Kapcsolódó tájékoztatás:

IBM Directory Server

RADIUS beállítása a syslogd démon használatára:

Ha a SYSLOG-ot szeretné használni a tevékenység- és a hibainformációk megjelenítésére, akkor engedélyeznie kell a syslogd démont.

A syslogd démon engedélyezéséhez végezze el az alábbi lépéseket.

1. Módosítsa az /etc/syslog.conf fájlt, és adja hozzá a következő bejegyzést: local4.debug var/adm/ipsec.log. A *local4* szolgáltatás használható a forgalom és az IP biztonsági események feljegyzésére. A naplózásra az operációs rendszer szabványos prioritásai vonatkoznak. A prioritási szintet érdemes mindaddig a *hibakeresés* szinten tartani, amíg az IP biztonsági alagutakon és szűrőkön áthaladó forgalom nem stabilizálódik.

   Megjegyzés: A szűrőesemények naplózása jelentős terhelést jelenthet az IP biztonsági hoszton, és nagy területet foglalhat el.
2. Mentse el a /etc/syslog.conf file fájlt.
3. Lépjen be a naplófájl számára megadott könyvtárba, és hozzon létre egy üres fájlt a megadott néven. A fenti példánál maradva lépjen be a /var/adm könyvtárba, és adja ki a **touch** parancsot a következőképpen:
   ```
   touch ipsec.log
   ```
4. Futtassa a **refresh** parancsot a syslogd alrendszeren a következőképpen:
   ```
   refresh -s syslogd
   ```

SYSLOG kimenet beállításainak megadása:

Megadhatja a 0, 3 vagy 9 Debug_Level-t, vagy beállíthatja a radiusd.conf fájlban, attól függően, hogy mennyi hibakeresési információra van szüksége a SYSLOG kimenetben.

Az alapértelmezett szint a 3-as. A radiusd.conf fájl hibakeresési része az alábbihoz hasonló:

```
#.
#.
#.
#   Debug_Level      :  This pair sets the debug level at which    #
#                       the RADIUS server will run.  Appropriate   #
#                       values are 0,3 or 9. The default is 3.     #
#                       Output is directed to location specified   #
#                       by *.debug stanza in /etc/syslog.conf      #
#                                                                  #
#                       Each level increases the amount of messages#
#                       sent to syslog. For example "9" includes   #
#                       the new messages provided by "9" as well   #
#                       as all messages generated by level 0 and 3.#
#                                                                  #
#                       0 : provides the minimal output to the     #
#                           syslogd log.  It sends start up        #
#                           and end messages for each RADIUS       #
#                           process.  It also logs error           #
#                           conditions.                            #
#                                                                  #
#                       3 : includes general ACCESS ACCEPT, REJECT #
#                           and DISCARD messages for each packet.  #
#                           This level provides a general audit    #
#                           trail for authentication.              #
#                                                                  #
#                       9 : Maximum amount of log data. Specific   #
#                           values of attributes while a           #
#                           transaction is passing thru            #
#                           processing and more.                   #
#                           [NOT advised under normal operations]  #
#                                                                  #
#------------------------------------------------------------------#
```

Az alábbiakban példa kimeneteket talál a különböző hibakeresési szintekhez.

Számlázási csomag 3-as hibakeresési szinten

```
Aug 18 10:23:57 server1 syslog: [0]:Monitor process [389288] has started
Aug 18 10:23:57 server1 radiusd[389288]: [0]:Local database (AVL) built.
Aug 18 10:23:57 server1 radiusd[389288]: [0]:Authentication process started : Pid= 549082 Port = 1812
Aug 18 10:23:57 server1 radiusd[389288]: [0]:Accounting process started : Pid= 643188 Port = 1813
Aug 18 10:23:57 server1 radiusd[643188]: [0]:Socket created [15]
Aug 18 10:23:57 server1 radiusd[643188]: [0]:Bound Accounting socket [15]
Aug 18 10:23:57 server1 radiusd[549082]: [0]:Socket created [15]
Aug 18 10:23:57 server1 radiusd[549082]: [0]:Bound Authentication socket [15]
Aug 18 10:24:07 server1 radiusd[643188]: [1]:*** Start Process_Packet() ***
Aug 18 10:24:07 server1 radiusd[643188]: [1]:Code 4, ID = 96, Port = 41639 Host = 10.10.10.10
Aug 18 10:24:07 server1 radiusd[643188]: [1]:ACCOUNTING-START - sending Accounting Ack to User [ user_id1 ]
Aug 18 10:24:07 server1 radiusd[643188]: [1]:Sending Accounting Ack of id 96 to 10.10.10.10 (client1.ibm.com)
Aug 18 10:24:07 server1 radiusd[643188]: [1]:send_acct_reply() Outgoing Packet:
Aug 18 10:24:07 server1 radiusd[643188]: [1]: Code = 5, Id = 96, Length = 20
Aug 18 10:24:07 server1 radiusd[643188]: [1]:*** Leave Process_Packet() ***
Aug 18 10:24:13 server1 radiusd[643188]: [2]:*** Start Process_Packet() ***
Aug 18 10:24:13 server1 radiusd[643188]: [2]:Code 4, ID = 97, Port = 41639 Host = 10.10.10.10
Aug 18 10:24:13 server1 radiusd[643188]: [2]:ACCOUNTING-STOP - sending Accounting Ack to User [ user_id1 ]
Aug 18 10:24:14 server1 radiusd[643188]: [2]:Sending Accounting Ack of id 97 to 10.10.10.10 (client1.ibm.com)
Aug 18 10:24:14 server1 radiusd[643188]: [2]:send_acct_reply() Outgoing Packet:
Aug 18 10:24:14 server1 radiusd[643188]: [2]: Code = 5, Id = 97, Length = 20
Aug 18 10:24:14 server1 radiusd[643188]: [2]:*** Leave Process_Packet() **
```

Számlázási csomag 9-es hibakeresési szinten

```
Aug 18 10:21:18 server1 syslog: [0]:Monitor process [643170] has started
Aug 18 10:21:18 server1 radiusd[643170]: [0]:Local database (AVL) built.
Aug 18 10:21:18 server1 radiusd[643170]: [0]:Authentication process started : Pid= 389284 Port = 1812
Aug 18 10:21:18 server1 radiusd[643170]: [0]:Accounting process started : Pid= 549078 Port = 1813
Aug 18 10:22:03 server1 radiusd[643170]: [0]:PID = [389284] dead
Aug 18 10:22:03 server1 radiusd[643170]: [0]:PID = [549078] dead
Aug 18 10:22:03 server1 radiusd[643170]: [0]:All child processes stopped. radiusd parent stopping
Aug 18 10:22:09 server1 syslog: [0]:Monitor process [1081472] has started
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Local database (AVL) built.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Inside client_init()
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Number of client entries read: 1
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Inside read_authorize_policy routine for file:
  /etc/radius/authorization/default.policy.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Inside read_authorize_file routine for file:
  /etc/radius/authorization/default.policy.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:read_authorize_file() routine complete.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Inside read_authorize_file routine for file:
  /etc/radius/authorization/default.auth.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:read_authorize_file() routine complete.
Aug 18 10:22:09 server1 radiusd[549080]: [0]:connect_to_LDAP_server:Database Location (where the data
  resides)=LDAP.
Aug 18 10:22:09 server1 radiusd[549080]: [0]:connect_to_LDAP_server:LDAP Server name= server1.austin.ibm.com.
Aug 18 10:22:09 server1 radiusd[549080]: [0]:connect_to_LDAP_server:LDAP Server port= 389.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Authentication process started : Pid= 549080 Port = 1812
Aug 18 10:22:09 server1 radiusd[389286]: [0]:connect_to_LDAP_server:Database Location (where the data
 resides)=LDAP.
Aug 18 10:22:09 server1 radiusd[389286]: [0]:connect_to_LDAP_server:LDAP Server name= server1.austin.ibm.com.
Aug 18 10:22:09 server1 radiusd[389286]: [0]:connect_to_LDAP_server:LDAP Server port= 389.
Aug 18 10:22:09 server1 radiusd[1081472]: [0]:Accounting process started : Pid= 389286 Port = 1813
Aug 18 10:22:10 server1 radiusd[549080]: [0]:Socket created [15]
Aug 18 10:22:10 server1 radiusd[549080]: [0]:Bound Authentication socket [15]
Aug 18 10:22:10 server1 radiusd[389286]: [0]:Socket created [15]
Aug 18 10:22:10 server1 radiusd[389286]: [0]:Bound Accounting socket [15]
Aug 18 10:22:15 server1 radiusd[389286]: [1]:*** Start Process_Packet() ***
Aug 18 10:22:15 server1 radiusd[389286]: [1]:Incoming Packet:
Aug 18 10:22:15 server1 radiusd[389286]: [1]:  Code = 4, Id = 94, Length = 80
Aug 18 10:22:15 server1 radiusd[389286]: [1]:  Authenticator = 0xC5DBDDFE6EFFFDBD6AE64CA35947DD0F
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =  40, Length =   6, Value = 0x00000001
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =   1, Length =   8, Value = 0x67656E747931
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =   4, Length =   6, Value = 0x00000000
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =   8, Length =   6, Value = 0x0A0A0A01
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =  44, Length =   8, Value = 0x303030303062
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =  30, Length =  10, Value = 0x3132332D34353638
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =  31, Length =  10, Value = 0x3435362D31323335
Aug 18 10:22:15 server1 radiusd[389286]: [1]:    Type =  85, Length =   6, Value = 0x00000259
```

```
Aug 18 10:22:15 server1 radiusd[389286]: [1]:Starting parse_packet()
Aug 18 10:22:15 server1 radiusd[389286]: [1]:Code 4, ID = 94, Port = 41639 Host = 10.10.10.10
Aug 18 10:22:15 server1 radiusd[389286]: [1]:Acct-Status-Type = Sta
```

Hitelesítési csomag 0-ás szinten

```
Aug 18 10:06:11 server1 syslog: [0]:Monitor process [1081460] has started
Aug 18 10:06:11 server1 radiusd[1081460]: [0]:Local database (AVL) built.
Aug 18 10:06:11 server1 radiusd[1081460]: [0]:Authentication process started : Pid= 549076 Port = 1812
Aug 18 10:06:11 server1 radiusd[1081460]: [0]:Accounting process started : Pid= 389282 Port = 18
```

Hitelesítési csomag 3-as szinten

```
Aug 18 10:01:32 server2 radiusd[389276]: [3]:*** Start Process_Packet() ***
Aug 18 10:01:32 server2 radiusd[389276]: [3]:Code 1, ID = 72, Port = 41638 Host = 10.10.10.10
Aug 18 10:01:32 server2 radiusd[389276]: [3]:authenticate_password_PAP: Passwords do not match, user is rejected
Aug 18 10:01:32 server2 radiusd[389276]: [3]:Authentication failed for user [user_id1] using IP [10.10.10.10]
Aug 18 10:01:32 server2 radiusd[389276]: [3]:ACCESS-REJECT - sending reject for id 72 to 10.10.10.10
  (client1.ibm.com)
Aug 18 10:01:32 server2 radiusd[389276]: [3]:send_reject() Outgoing Packet:
Aug 18 10:01:32 server2 radiusd[389276]: [3]: Code = 3, Id = 72, Length = 30
Aug 18 10:01:32 server2 radiusd[389276]: [3]:*** Leave Process_Packet() ***
Aug 18 10:01:53 server2 radiusd[389276]: [4]:*** Start Process_Packet() ***
Aug 18 10:01:53 server2 radiusd[389276]: [4]:Code 1, ID = 74, Port = 41638 Host = 10.10.10.10
Aug 18 10:01:53 server2 radiusd[389276]: [4]:authenticate_password_PAP: Passwords Match, user is authenticated
Aug 18 10:01:53 server2 radiusd[389276]: [4]:Authentication successful for user [user_id1] using IP [10.10.10.10]
Aug 18 10:01:53 server2 radiusd[389276]: [4]:Authorization successful for user [user_id1] using IP [10.10.10.10]
Aug 18 10:01:53 server2 radiusd[389276]: [4]:ACCESS-ACCEPT - sending accept for id 74 to 10.10.10.10
  (client1.ibm.com)
Aug 18 10:01:53 server2 radiusd[389276]: [4]:send_accept() Outgoing Packet:
Aug 18 10:01:53 server2 radiusd[389276]: [4]: Code = 2, Id = 74, Length = 31
Aug 18 10:01:53 server2 radiusd[389276]: [4]:*** Leave Process_Packet() **
```

Hitelesítési csomag 9-es szinten

```
Aug 18 10:03:56 server1 radiusd[389278]: [1]:*** Start Process_Packet() ***
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Incoming Packet:
Aug 18 10:03:56 server1 radiusd[389278]: [1]:  Code = 1, Id = 77, Length = 58
Aug 18 10:03:56 server1 radiusd[389278]: [1]:  Authenticator = 0xE6CB0F9C22BB4E799854E734104FB2D5
Aug 18 10:03:56 server1 radiusd[389278]: [1]:    Type =   1, Length =   8, Value = 0x67656E747931
Aug 18 10:03:56 server1 radiusd[389278]: [1]:    Type =   4, Length =   6, Value = 0x00000000
Aug 18 10:03:56 server1 radiusd[389278]: [1]:    Type =   2, Length =  18, Value = 0x**********
**********************
Aug 18 10:03:56 server1 radiusd[389278]: [1]:    Type =   7, Length =   6, Value = 0x00000001
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Starting parse_packet()
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Code 1, ID = 77, Port = 41638 Host = 10.10.10.10
Aug 18 10:03:56 server1 radiusd[389278]: [1]:User-Name = "user_id1"
Aug 18 10:03:56 server1 radiusd[389278]: [1]:NAS-IP-Address = 10.10.10.10
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Framed-Protocol = PPP
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Leaving parse_packet()
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Verifying Message-Authenticator
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Message-Authenticator successfully verified
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside proxy_request_needed() function
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Proxy is not turned on
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Username = [user_id1]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Client IP = [10.10.10.10]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside parse_for_login( user_id1 )
Aug 18 10:03:56 server1 radiusd[389278]: [1]:User_id remaining after prefix removal = [user_id1]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:User_id remaining after suffix removal = [user_id1]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside rad_authenticate() function
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Authentication request received for [client1.austin.ibm.com]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Calling get_ldap_user() to get LDAP user data
Aug 18 10:03:56 server1 radiusd[389278]: [1]:get_ldap_user:LDAP user id: user_id1.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:get_ldap_user:LDAP max_login_cnt:2.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:get_ldap_user:LDAP EAP_type: 4.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:get_ldap_user:LDAP passwordexpiredweeks: 9.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:get_ldap_active_sessions:number of free entries= 2.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:get_ldap_active_session:dn retrieved=
radiusuniqueidentifier=user_id11,ou=radiusActiveUsers,cn=aixradius.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside get_client_secret routine for ip:10.10.10.10
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Found NAS-IP = [10.10.10.10]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Found shared secret.
```

```
Aug 18 10:03:56 server1 radiusd[389278]: [1]:authenticate_password_PAP: Passwords Match, user is authenticated
Aug 18 10:03:56 server1 radiusd[389278]: [1]:is_ldap_pw:password for user has NOT expired
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Authentication successful for user [user_id1] using IP [10.10.10.10]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside rad_authorize() routine.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside read_authorize_policy routine for file:
  /etc/radius/authorization/user_id1.policy.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside read_authorize_file routine for file:
  /etc/radius/authorization/user_id1.policy.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Did not open /etc/radius/authorization/user_id1.policy file.
  File may not be found.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Error reading policy file: /etc/radius/authorization/user_id1.policy.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:rad_authorize:default policy list and userpolicy list were empty.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:In create_def_copy() routine.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Successfully made a copy of the master authorization list.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside read_authorize_file routine for file:
  /etc/radius/authorization/user_id1.auth.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Did not open /etc/radius/authorization/user_id1.auth file.
  File may not be found.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:rad_authorize:copy authorization list and user list were empty.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Authorization successful for user [user_id1] using IP [10.10.10.10]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:ACCESS-ACCEPT - sending accept for id 77 to 10.10.10.10
  (client1.austin.ibm.com)
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside proxy_response_needed() function
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Proxy is not turned on
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Inside get_client_secret routine for ip:10.10.10.10
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Found NAS-IP = [10.10.10.10]
Aug 18 10:03:56 server1 radiusd[389278]: [1]:Found shared secret.
Aug 18 10:03:56 server1 radiusd[389278]: [1]:send_accept() Outgoing Packet:
Aug 18 10:03:56 server1 radiusd[389278]: [1]:  Code = 2, Id = 77, Length = 31
Aug 18 10:03:56 server1 radiusd[389278]: [1]:send_accept() Outgoing Packet:
Aug 18 10:03:56 server1 radiusd[389278]: [1]:  Code = 2, Id = 77, Length = 31
Aug 18 10:03:56 server1 radiusd[389278]: [1]:  Authenticator = 0xCCB2B645BBEE86F5E4FC5BE24E904B2A
Aug 18 10:03:56 server1 radiusd[389278]: [1]:    Type =  18, Length =  11, Value = 0x476F6F646E65737321
Aug 18 10:03:56 server1 radiusd[389278]: [1]:*** Leave Process_Packet() ***
Aug 18 10:04:18 server1 radiusd[389278]: [2]:*** Start Process_Packet() ***
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Incoming Packet:
Aug 18 10:04:18 server1 radiusd[389278]: [2]:  Code = 1, Id = 79, Length = 58
Aug 18 10:04:18 server1 radiusd[389278]: [2]:  Authenticator = 0x774298A2B6DD90D7C33B3C10C4787D41
Aug 18 10:04:18 server1 radiusd[389278]: [2]:    Type =   1, Length =   8, Value = 0x67656E747931
Aug 18 10:04:18 server1 radiusd[389278]: [2]:    Type =   4, Length =   6, Value = 0x00000000
Aug 18 10:04:18 server1 radiusd[389278]: [2]:    Type =   2, Length =  18, Value = 0x*******
************************
Aug 18 10:04:18 server1 radiusd[389278]: [2]:    Type =   7, Length =   6, Value = 0x00000001
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Starting parse_packet()
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Code 1, ID = 79, Port = 41638 Host = 10.10.10.10
Aug 18 10:04:18 server1 radiusd[389278]: [2]:User-Name = "user_id1"
Aug 18 10:04:18 server1 radiusd[389278]: [2]:NAS-IP-Address = 10.10.10.10
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Framed-Protocol = PPP
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Leaving parse_packet()
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Verifying Message-Authenticator
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Message-Authenticator successfully verified
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Inside proxy_request_needed() function
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Proxy is not turned on
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Username = [user_id1]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Client IP = [10.10.10.10]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Inside parse_for_login( user_id1 )
Aug 18 10:04:18 server1 radiusd[389278]: [2]:User_id remaining after prefix removal = [user_id1]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:User_id remaining after suffix removal = [user_id1]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Inside rad_authenticate() function
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Authentication request received for [client1.austin.ibm.com]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Calling get_ldap_user() to get LDAP user data
Aug 18 10:04:18 server1 radiusd[389278]: [2]:get_ldap_user:LDAP user id: user_id1.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:get_ldap_user:LDAP max_login_cnt:2.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:get_ldap_user:LDAP EAP_type: 4.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:get_ldap_user:LDAP passwordexpiredweeks: 9.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:get_ldap_active_sessions:number of free entries= 2.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:get_ldap_active_session:dn retrieved=
  radiusuniqueidentifier=user_id11, ou=radiusActiveUsers, cn=aixradius.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Inside get_client_secret routine for ip:10.10.10.10
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Found NAS-IP = [10.10.10.10]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Found shared secret.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:authenticate_password_PAP: Passwords do not match, user is rejected
```

```
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Authentication failed for user [user_id1] using IP [10.10.10.10]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:ACCESS-REJECT - sending reject for id 79 to 10.10.10.10
 (client1.austin.ibm.com)
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Inside proxy_response_needed() function
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Proxy is not turned on
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Inside get_client_secret routine for ip:10.10.10.10
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Found NAS-IP = [10.10.10.10]
Aug 18 10:04:18 server1 radiusd[389278]: [2]:Found shared secret.
Aug 18 10:04:18 server1 radiusd[389278]: [2]:send_reject() Outgoing Packet:
Aug 18 10:04:18 server1 radiusd[389278]: [2]:  Code = 3, Id = 79, Length = 30
Aug 18 10:04:18 server1 radiusd[389278]: [2]:send_reject() Outgoing Packet:
Aug 18 10:04:18 server1 radiusd[389278]: [2]:  Code = 3, Id = 79, Length = 30
Aug 18 10:04:18 server1 radiusd[389278]: [2]:  Authenticator = 0x05D4865C6EBEFC1A9300D2DC66F3DBE9
Aug 18 10:04:18 server1 radiusd[389278]: [2]:    Type =  18, Length =  10, Value = 0x4261646E65737321
Aug 18 10:04:18 server1 radiusd[389278]: [2]:*** Leave Process_Packet() **
```

Jelszó lejárata

A jelszólejárat segítségével a RADIUS kliens értesítést kap arról, ha a felhasználói jelszó lejárt, és ez a jelszó RADIUS protokollon keresztül frissíthető.

A jelszó lejárat szolgáltatás négy új csomagtípust és egy új attribútumot érint. Az új csomagtípusokat az AIX szótár tartalmazza, csak be kell kapcsolni a szolgáltatást.

Nem minden RADIUS környezetben kívánatos a lejárt jelszavak RADIUS protokollon keresztüli frissítése. A lejárt jelszavak RADIUS protokollon keresztüli módosítását a radiusd.conf fájlban lehet engedélyezni vagy letiltani. A funkció alapértelmezésben le van tiltva. Felveheti egy Password_Expired_Reply_Message felhasználói válaszüzenetet, amely a password-expired csomagban kerül visszaadásra. Az új és régi jelszó attribútumokat egyaránt szükséges titkosítani illetve visszafejteni a PAP módszer segítségével.

Szállító saját attribútumai

A szállító saját attribútumait (VSA) a távoli elérés szerver szállítója, általában a hardverszállító határozza meg a RADIUS működésének testreszabásához az adott hardveren.

A szállító saját attribútumai akkor szükségesek, ha a felhasználók számára többféle elérést is engedélyezni kíván. A VSA-k a RADIUS által meghatározott attribútumokkal együtt használhatók.

A szállító saját attribútumainak használata nem kötelező, azonban ha a NAS hardver a megfelelő működéshez további konfigurációt igényel, akkor fel kell venni ezeket az attribútumokat a szótárfájlba.

A **User-Name** és **Password** attribútumokkal együtt a szállító saját attribútumait a felhatalmazás kibővítéséhez is fel lehet használni. A szerveroldalon a felhasználói felhatalmazás házirend fájl tartalmazza azokat az attribútumokat, amelyeket egy adott felhasználótól érkező hozzáférés-kérés csomagban ellenőrizni kell. Ha a csomag nem tartalmazza a fájlban felsorolt attribútumokat, akkor a RADIUS a NAS szervernek hozzáférés_visszautasítva csomagot küld. A szállító saját attribútumait attribútum=érték párok listájaként is meg lehet adni a ***felhasználói_azonosító***.policy fájlban.

Az alábbi részlet egy minta dictionary fájlból származik, és a szállító saját attribútumainak használatát mutatja be:

```
########################################################################
#                                                                      #
#   This section contains examples of dictionary translations for      #
#   parsing vendor specific attributes (vsa). The example below is for #
#   "Cisco." Before defining an Attribute/Value pair for a             #
#   vendor a "VENDOR" definition is needed.                            #
#                                                                      #
#   Example:                                                           #
#                                                                      #
#   VENDOR          Cisco           9                                  #
#                                                                      #
#   VENDOR:  This specifies that the Attributes after this entry are   #
#            specific to Cisco.                                        #
#   Cisco :  Denotes the Vendor name                                   #
#   9     :  Vendor Id defined in the "Assigned Numbers" RFC           #
#                                                                      #
```

```
#########################################################################

#VENDOR          Cisco           9

#ATTRIBUTE       Cisco-AVPair              1       string
#ATTRIBUTE       Cisco-NAS-Port            2       string
#ATTRIBUTE       Cisco-Disconnect-Cause    195     integer
#
#----------------Cisco-Disconnect-Cause---------------------------------#
#
#VALUE           Cisco-Disconnect-Cause    Unknown                 2
#VALUE           Cisco-Disconnect-Cause    CLID-Authentication-Failure 4
#VALUE           Cisco-Disconnect-Cause    No-Carrier              10
#VALUE           Cisco-Disconnect-Cause    Lost-Carrier            11
#VALUE           Cisco-Disconnect-Cause    No-Detected-Result-Codes    12
#VALUE           Cisco-Disconnect-Cause    User-Ends-Session       20
#VALUE           Cisco-Disconnect-Cause    Idle-Timeout            21
#VALUE           Cisco-Disconnect-Cause    Exit-Telnet-Session     22
#VALUE           Cisco-Disconnect-Cause    No-Remote-IP-Addr       23
```

RADIUS válaszüzenet támogatás

A válaszüzenet a radiusd.conf fájlban létrehozott és beállított szöveg.

A válaszüzenetek célja az, hogy a NAS vagy AP szövegesen tájékoztathassa a felhasználót a művelet eredményéről, ami lehet siker, sikertelenség vagy felhívás. A válaszüzenetek olvasható szöveges mezők, melyek tartalma megvalósításfüggő, és beállításuk a szerver konfigurálásakor történik. Az attribútumok alapérték üres szöveg. Az attribútumok közül bármelyiket beállíthatja vagy üresen hagyhatja a többitől függetlenül.

A RADIUS az alábbi válaszüzenet attribútumokat támogatja:

- Elfogadva válaszüzenet
- Visszautasítva válaszüzenet
- CHAP válaszüzenet
- Jelszó lejárt válaszüzenet

Az attribútumokat a démon indításkor a radiusd.conf konfigurációs fájlból a globális konfigurációs struktúrába olvassa be. Az értékeket a SMIT **Szerver beállítása** menüpontjához tartozó RADIUS képernyőkön állíthatja be. Az egyes karakterláncok legfeljebb 256 karakter hosszúak lehetnek.

A funkció az alábbiak szerint van megvalósítva:

1. A **radiusd** démon indításkor beolvassa a radiusd.conf fájlt, és beállítja a válaszüzenet attribútumokat.
2. Ha hozzáférés kérés csomag érkezik, akkor a felhasználó hitelesítésre kerül.
3. Ha a hitelesítési válasz hozzáférés engedélyezve, akkor az Elfogadva válaszüzenet szövege ellenőrzésre kerül. Ha a szöveg be van állítva, akkor a karakterlánc visszaküldésre kerül a hozzáférés engedélyezve csomagban.
4. Ha a hitelesítés sikertelen, akkor a Visszautasítva válaszüzenetet szövege kerül ellenőrzésre, és visszaküldésre kerül a hozzáférés elutasítva csomagban.
5. Ha a hitelesítés felhívás-válasz típusú, akkor a CHAP válaszüzenet attribútum ellenőrzésre és elküldésre kerül a Hozzáférés-Felhívás csomag részeként.

RADIUS szerver IP tároló konfigurációja

A RADIUS szerver segítségével dinamikusan hozzárendelhet egy IP címet az IP címtárolóból.

Az IP cím kiosztása a felhatalmazási folyamat része és a hitelesítés után kerül végrehajtásra. A rendszeradminisztrátornak egy egyedi IP címet kell hozzárendelnie a felhasználókhoz. Az IP cím dinamikus biztosításához a RADIUS szerver három lehetőséget kínál:

- Framed Pool attribútum
- A Vendor Specific attribútum használata

- RADIUS szerveroldali IP tárolás

Framed Pool attribútum

A *tárolónév* IP tárolót a Network Access Serveren (NAS) kell megadni. A NAS-nak meg kell felelnie a RFC2869 szabványnak, hogy a RADIUS szerver el tudjon küldeni egy **Framed-Pool** attribútumot egy Hozzáférés-elfogadás csomagban (type 88 attribútum). A rendszeradminisztrátornak be kell állítania a NAS-t és frissítenie kell a felhasználó felhatalmazási attribútumait az alábbi módon: megadja a **Framed-Pool** attribútumot a globális default.auth fájlban vagy a user.auth fájlban a RADIUS szerveren. A RADIUS szerveren lévő szótárfájl tartalmazza ezt az attribútumot:

```
ATTRIBUTE    Framed-Pool    88       string
```

Ha a NAS nem tud több címtárolót használni, akkor figyelmen kívül hagyja ezt az attribútumot. A NAS címtárolója IP címek listáját tartalmazza. A NAS dinamikusan felveszi a megadott tárolóban meghatározott egyik IP címet és hozzárendeli a felhasználóhoz.

Szállító-specifikus attribútumok

Néhány független szoftverszállító (ISV) nem tudja használni a **Framed-Pool** attribútumot, de meg tud adni IP címtárolót. A RADIUS szerver a Szállító-specifikus attribútum (VSA) modell segítségével ki tudja használni ezeket a címtárolókat. A Cisco NAS például egy Cisco-AVPair nevű attribútumot biztosít. A RADIUS szerveren lévő szótárfájl tartalmazza ezt az attribútumot:

```
VENDOR     Cisco          9
ATTRIBUTE  Cisco-AVPair   1       string
```

Amikor a NAS egy Hozzáférés kérés csomagot küld, az tartalmazza ezt az Cisco-AVPair=”ip:addr-pool=*tárolónév*” attribútumot, ahol a *tárolónév* a NAS-on megadott címtároló neve. A kérés hitelesítése és felhatalmazása után a RADIUS szerver visszaküldi az attribútumot a Hozzáférés elfogadása csomagban. A NAS ezután használhatja a megadott tárolót az IP cím felhasználó számára kiosztásához. A rendszeradminisztrátornak be kell állítania a NAS-t és frissítenie kell a felhasználó felhatalmazási attribútumait az alábbi módon: megadja a VAS attribútumot a globális default.auth fájlban vagy a RADIUS szerveren lévő user.auth fájlban.

Radius szerveroldali IP tárolás

A RADIUS szerver beállítható úgy, hogy az IP címek tárolójából állítson elő egy IP címet. Az IP címe a Hozzáférés elfogadás csomag Framed-IP-Address attribútumában kerül visszaküldésre.

A rendszeradminisztrátor az SMIT felület segítségével megadhat egy IP címtárolót. A címeket az /etc/radius/ippool_def fájl tárolja. A *tárolónevek* az etc/radius/clients fájlban vannak megadva. A rendszeradminisztrátornak a NAS portszámot is be kell állítania. A RADIUS szerver démon az etc/radius/clients és /etc/radius/ippool_def fájl információit használja az adatfájlok létrehozásához. A démon elindulása után a rendszeradminisztrátor nem módosíthatja és nem vehet fel *tárolóneveket* vagy IP címtartományokat, amíg a RADIUS szerverek leállításra nem kerülnek. A RADIUS szerver démon indításkor beolvassa a konfigurációs fájlt (/etc/radius/radius.conf) és ha az IP kiosztás engedélyezve van (Enable_IP_Pooling=YES), akkor On értéket ad a globális IP kiosztás jelzőnek (IP_pool_flag). A démon ezután ellenőrzi, hogy a poolname.data fájl létezik-e. Ha létezik, akkor beolvassa a fájlt és az információkat az elosztott memóriában eltárolja. Ezután a kliensektől érkező kérések alapján frissíti a fájlt és az elosztott memóriát. Ha a fájl nem létezik, akkor a démon az etc/radius/clients és a /etc/radius/ippool_def fájlban lévő információk segítségével létrehoz egy új fájlt. A poolname.data fájl maximális mérete 256 MB (AIX szegmensméretkorlát). Ha a poolname.data fájl nagyobb, mint 256 MB, akkor a RADIUS szerver naplózza a hibaüzenetet és kilép.

A démon az /etc/radius/ippool_def fájlból IP tároló részleteket kér le és az elosztott memóriában minden tárolónévhez fenntart egy táblázatot az IP címekkel. A táblázatban NAS-IP-address, NAS-port és IN USE jelző bejegyzések találhatók. A démon fenntart egy kivonattáblát, amelynek a kulcsa a NAS-IP NAS port. Ha több felhasználótól érkezik kérés, akkor az UDP sorba rakja a kéréseket, és a démona a kérésből lékéri a NAS-IP és NAS port adatokat. Ezen információk segítségével ellenőrzi, hogy a *tárolónév* meg lett-e adva a NAS-hoz etc/radius/clients fájl információinak kiolvasásával.

A démon megpróbál a tárolóból egy nem használt címet lekérni. Ha van nem használt cím, akkor a NAS-IP és NAS-port jelző “használatban van” jelzővel látja el, és ez visszaküldésre kerül a RADIUS szerverhez. A démon az IP címet beteszi a **Framed-IP-Address** attribútumba, és az elfogadás csomagban visszaküldi a NAS-nak. A poolname.data fájl szintén frissítésre kerül, hogy szinkronban legyen az elosztott memóriában lévő információkkal.

Ha a tároló nem létezik, vagy létezik de nem tartalmaz több nem használt címet, akkor a RADIUS szerver hibát kap. A Nem osztható ki IP cím hiba bekerül a naplófájlba és a RADIUS szerver egy Hozzáférés visszautasítva csomagot küld a NAS-nak.

Hibakódok:

- NOT_POOLED – A **nas_ip**-hez nincs megadva tároló.
- POOL_EXHAUSTED – A **nas_ip**-hez meg van adva tároló, de a benne lévő címek már használatban vannak.

Amikor a hitelesítési kérés megérkezik egy NAS-ról, és a NAS port kombináció már rendelkezik egy kiosztott IP-címmel, a démon visszaadja az előző kiosztott címet tárolónak azzal, hogy az IN USE jelzőt Off értékre állítja, valamint törli a NAS-IP-cím és NAS-port bejegyzéseket a táblázatban. Ezután új IP-címet oszt ki a tárolóból.

Az IP-cím szintén visszaküldésre kerül a tárolóba, amikor a RADIUS szerver Számlázás leállítása csomagot kap a NAS-tól. A Számlázás leállítása csomagnak tartalmaznia kell a NAS-IP-cím és a NAS-port megjegyzéseket. A démon a következő esetekben hozzáfér az ippool_mem fájlhoz:

- Egy új IP cím kérésére szolgáló kérés érkezik. A HASZNÁLATBAN jelzőt igazra állítja.
- Egy Accounting-Stop csomag érkezik. Ez felszabadítja az IP címet a “használatban” jelző false-ra állításával.

Minden esetében a megosztott memória rendszerhívások biztosítják, hogy a megosztott memóriában lévő adatok és a poolname.data fájlok szinkronban legyenek. A rendszeradminisztátor az IP kiosztást be- (ON) vagy kikapcsolhatja (OFF) a RADIUS szerver konfigurációs fájl (radiusd.conf) *Enable_IP_Pooling* paramétere segítségével. Ez akkor hasznos, ha a rendszeradminisztrátor egy hozzárendelt IP címmel rendelkezik a globális default.auth vagy user.auth fájlban. Hozzárendelt IP cím használatához a rendszeradminisztrátornak be kell állítani az Enable_IP_Pool = NO értéket.

A /etc/radius/ippool_def fájl példája SMIT segítségével került létrehozásra:

Tárolónév	Tartomány kezdete	Tartomány vége
Floor5	192.165.1.1	192.165.1.125
Floor6	192.165.1.200	192.165.1.253

Az alábbi /etc/radiusclients fájl SMIT segítségével került létrehozásra:

NAS-IP	Osztott titok	Tárolónév
1.2.3.4	Secret1	Floor5
1.2.3.5	Secret2	Floor6
1.2.3.6	Secret3	Floor5
1.2.3.7	Secret4	

A fenti 1.2.3.7 NAS-IP cím példában a tárolónév üres. Ebben az esetben az IP tárkezelés nem zajlik le ehhez a NAS-hoz (még akkor se, ha a global IP_pool_flag = True). Amikor a hozzáféréskérési (Access-Request) csomag megérkezik, a RADIUS szerver elvégzi a hitelesítést és a jogosultságok megadását. Ha sikerrel jár, elküldi a kérésben meghatározott, a globális default.auth fájlból vagy a user.auth fájlból származó IP-címet az Access-Accept (kéréselfogadási) csomagban. Ebben az esetben nincs szükség a NAS-Port attribútumra.

Ha az IP-tárolókezelés értéke True, a rendszeradminisztrátor megadott IP-címet is a globális default.auth vagy user.auth részeként vagy az Access-Request csomag részeként. A RADIUS szerver behelyettesíti ezt az IP-címet az adott NAS-hoz meghatározott nevű tárolóból kiosztott címmel. Ha a tároló összes IP-címe foglalt, a szerver naplózza a hibát (tároló megtelt), és egy Access-Reject (hozzáférés elutasítva) csomagot küld. A szerver figyelmen kívül hagy minden, az auth fájlokban megadott IP-címet.

Ha az IP-tárkezelés értéke True és érvényes tárolónév van megadva a NAS-hoz, akkor amikor egy Access-Request érkezik arról a NAS IP-címről, és nem rendelkezik meghatározott NAS porttal, a szerver egy Access-Reject csomagot küld.

A következő theFloor5.data fájl példa démonnal kerül létrehozásra:

IP cím	NAS-IP	NAS-Port	Használatban
192.165.1.1	1.2.3.4	2	1
192.165.1.2	1.2.3.4	3	0
.....	.....	....	....
192.165.1.124	1.2.3.6	1	1
192.165.1.125	1.2.3.6	6	1

A következő theFloor6.data fájl példa démonnal kerül létrehozásra:

IP cím	NAS-IP	NAS-Port	Használatban
192.165.200	1.2.3.4	1	1
192.165.201	1.2.3.4	4	1
.....	.....	....	....
192.165.1.252	1.2.3.4	5	0
192.165.1.253	1.2.3.4	6	1

Ha minden kiosztott IP-címet fel kell oldani egy adott NAS-hoz (például mert egy NAS leáll), szükség lehet az összes tárolóból származó összes IP-cím feloldására a *poolname*.data fájl inicializálásához. A rendszeradminisztrátor a következő menüműveleteket végezheti el a SMIT segítségével:

- IP-tároló törlése egy klienshez
- A teljes IP-tároló törlése

Az IP-tároló SMIT paneljei

A Klienskonfigurációban a **Kliens megadása** panelben megadhat egy szabadon választott **Tárolónevet**. A név maximum 64 karakterből állhat. Ha a **Tárolónév** üres, az IP tárolókezelés nem zajlott le és a RADIUS szerver a rendszeradminisztrátor által a **Framed-IP-Address** hitelesítési attribútumban megadott IP-címet rendeli hozzá.

Ha az **IP tároló** van kiválasztva, a következő beállítások jelennek meg:

- Minden IP-tároló listázása
- IP-tároló létrehozása
- IP-tároló jellemzőinek módosítása/megjelenítése
- IP-tároló törlése
- IP-tároló törlése egy klienshez
- A teljes IP-tároló törlése

Minden IP-tároló listázása: Ezzel a beállítással kilistázható a **Tárolónév**, az **IP-címek indulási tartománya** és az **IP-címek befejezési tartománya**.

IP-tároló létrehozása: Ezzel a beállítással adható meg a tároló neve, az indulási és a befejezési tartomány. Ezek az adatok az ippool_def fájl végéhez adódnak hozzá. A rendszer ellenőrzéseket végez annak biztosítására, hogy ne fordulhassanak elő ismétlődő tárolónevek és hogy az IP-címtartományok nem érnek össze. Ezt a műveletet csak akkor lehet végrehajtani, ha a RADIUS szerverdémonok nem futnak.

IP-tároló jellemzőinek módosítása/megjelenítése: Ez a beállítás megjeleníti a tárolónevek listáját egy előugró panelben. Ebből a panelből ki kell választania egy adott tárolónevet. Amikor ez megtörténik, megjelenik a választott névvel rendelkező panel. Ha Entert nyom, a tárolónév adatai frissítésre kerülnek az ippool_def fájlban. Ezt a műveletet csak akkor lehet végrehajtani, ha a RADIUS szerverdémonok nem futnak.

IP-tároló törlése: Ennek a lehetőségnek a kiválasztására megjelenik a kiválasztható tárolónevek listája. Ha kiválaszt egy nevet, akkor a **Bizonyos benne?** előugró kérdésnél meg kell erősítenie törlési szándékát; csak ezután törlődik a kiválasztott tárolónév. Meghívódik az **rmippool** parancsfájl a kiválasztott tárolónév törlésére az ippool_def fájlból. Ezt a műveletet csak akkor lehet végrehajtani, ha a RADIUS szerverdémonok nem futnak.

IP-tároló törlése egy klienshez: Ez a lehetőség 0 értékre állítja a NAS-hoz tartozó IP-címek **IN-USE** bejegyzését, ami azt jelenti, hogy most minden, a NAS-hoz tartozó IP-cím rendelkezésre áll. Ezt a művelet csak akkor történhet meg, ha a RADIUS szerverdémonok nem futnak.

A teljes IP-tároló törlése: Ennek a lehetőségnek a kiválasztásakor megjelenik egy **Bizonyos benne?** előugró ablak, ahol meg kell erősítenie szándékát a teljes ippool_mem fájl kiürítése előtt. Ezt a műveletet csak akkor lehet végrehajtani, ha a RADIUS szerverdémonok nem futnak.

RADIUS SMIT párbeszédablakok

Ha a RADIUS szervert SMIT segítségével állítja be, akkor a csillaggal (*) jelölt mezők kitöltése kötelező.

A SMIT gyorselérése:

```
smitty radius
```

A RADIUS főmenüje:

```
       RADIUS Server

Configure Server
Configure Clients
Configure Users
Configure Proxy Rules
Advanced Server Configuration
Start RADIUS Server daemons
Stop RADIUS Server daemons
```

Az alábbi képernyőfotó egy példa RADIUS Szerver beállítása SMIT párbeszédablak látható:

```
Configure Server
  RADIUS Directory            /etc/radius
* Database Location                         [Local]                  +
  Local AVL Database File Name [dbdata.bin]
  Debug Level                               [9]                      +#
  Local Accounting                          [ON]                     +
  Local Accounting Directory                [/var/radius/data/accou>
  Accept Reply-Message          []
  Reject Reply-Message          []
  Challenge Reply-Mesage                    []
  Password Expired Reply-Message            []
  Support Renewal of Expired Password       [NO]                     +
  Require Message Authenticator             [NO]                     +
*Authentication Port Number  [1812]
*Accounting Port Number      [1813]
  LDAP Server Name              []
  LDAP Server Port Number                   [389]                     #
  LDAP Server Admin Distinguished Name      [cn=root]
  LDAP Server Admin Password          []
  LDAP Base Distinguished Name      [cn=aixradius]
  LDAP Size Limit                           [0]                       #
  LDAP Hop Limit                            [0]                       #
  LDAP wait time limit                      [10]                      #
  LDAP debug level                          [0]                      +#
  Proxy Allowed                             [OFF]                    +
  Proxy Use Table                           [OFF]                    +
  Proxy Realm Name                  []
  Proxy Prefix Delimiters           [$/]
  Proxy Suffix Delimiters           [@.]
  Proxy Remove Hops                         [NO]                     +
  Proxy Retry Count                         [2]                       #
  Proxy Timeout                             [30]                      #
  UNIX Check Login Restrictions             [OFF]                    +
  Enable IP Pool                            [OFF]                    +
  Send Message Authenticator for ACCEPT     [ON]                     +
  Maximum RADIUS Server Threads             [15]                      #
  EAP Conversation Timeout (Seconds)        [30]                      #
  Enable EAP-TLS                            [ON]                     +
  Required Options for EAP-TLS
       Path to OpenSSL Library              [/opt/freeware/lib/libs>
       OpenSSL Cipher List                  [ALL:!ADH:RC4+RSA:+SSLv>
       Root CA Directory (Full Path)        [/etc/radius/tls]
       Root CA Certificate (Full Path)      [/etc/radius/tls/radius>
       RADIUS Server Certificate (Full Path) [/etc/radius/tls/cert-s>
       RADIUS Server Private Key (Full Path) [/etc/radius/tls/cert-s>
       RADIUS Server CRL (Full Path)        []
```

Az **F1** billentyű megnyomásával részletes SMIT súgóinformációkat kaphat az összes elérhető mezőről és menüpontról.

Véletlenszám generátor

A RADIUS csomag Authenticator mezőjének kitöltéséhez véletlenszámokra van szükség.

Fontos, hogy a lehető legjobb véletlenszám-generátor álljon rendelkezésre, mert ellenkező esetben egy támadó megpróbálhatja úgy megkerülni a hitelesítést, hogy elküld egy megjósolható kérést a RADIUS szervernek, majd a választ felhasználva egy későbbi hozzáférés kérésre válaszolhat úgy, mintha a válaszok a RADIUS szervertől érkeznének. Az AIX RADIUS szerver a **/dev/urandom** kernelbővítményt használja pszeudo véletlenszámok előállítására. Ez a modul entrópiamintákat gyűjt a hardveres forrásból egy ál-illesztőprogram útján. Az eszköz a véletlenszámok megfelelőségének biztosítása érdekében NIST tesztelésen esett át.

Nemzetközi támogatás

A RADIUS **raddbm** parancs és a SMIT képernyők a szabványos AIX globalizációs API hívásokon keresztül rendelkeznek nemzetközi támogatással.

Kapcsolódó információk

Parancsok: **installp**,**mkuser** és **raddbm**

AIX behatolásvédelem

Az AIX behatolásvédelem felismeri a nem megfelelő, jogosulatlan vagy más, a rendszerre esetleg károsan ható adatforgalmat.

Ez a fejezet az AIX által biztosított különböző behatolásvédelmi típusokkal foglalkozik.

Kapcsolódó információk

Parancsok: **chfilt**, **ckfilt**, **expfilt**, **genfilt**, **impfilt**, **lsfilt**, **mkfilt**, **mvfilt**, **rmfilt**.

Behatolásészlelés

A behatolásvédelem a rendszerműveletek megfigyelésének és elemzésének folyamata, amely a jogosulatlan rendszerelérésre tett kísérletek felismerésére és visszautasítására irányul. AIX rendszeren a jogosulatlan elérést illetve az erre tett kísérleteket bizonyos műveletek megfigyelésével és szűrőszabályok alkalmazásával valósítja meg.

Megjegyzés: A behatolásvédelem engedélyezéséhez telepítse a bos.net.ipsec fájlkészleteket a hoszt rendszerre. Az észlelési technológia a meglévő AIX Internet biztonsági protokoll (IPsec) szolgáltatásokon alapszik.

Mintaillesztési szűrőszabályok:

A mintaillesztés a hálózati csomagok szűrését jelenti IPSec szűrőszabályok alkalmazásával. A szűrőminta lehet egy szöveges vagy hexadecimális karakterlánc, illetve egy több mintát tartalmazó fájl. Ha az egyik szűrőszabály alapján azonosított minta felbukkan egy hálózati csomag törzsében, akkor a szűrőszabályhoz előre meghatározott művelet fut le.

A behatolásvédelem a mintaillesztési szűrőszabályokat csak a bejövő hálózati forgalomra alkalmazza. A szűrőszabály táblához a **genfilt** paranccsal adhat hozzá egy szűrőszabályt. A parancs által előállított szűrőszabályokat nevezzük kézi szűrőszabályoknak. Az **mkfilt** paranccsal aktiválhatja vagy leállíthatja a szűrőszabályok alkalmazását. Az **mkfilt** paranccsal a szűrőnaplózás funkció is vezérelhető.

A minta fájl soronként egy szöveges vagy hexadecimális mintát tartalmaz. A mintaillesztési szűrőszabályok használhatók vírusok valamint puffertúlcsordulásos és egyéb hálózati biztonsági támadások elhárítására.

Ha azonban széleskörűen és nagy számban alkalmazza a szűrőszabályokat, akkor az negatív hatással vannak a rendszer teljesítményére. A legjobb megoldás, ha a lehető legszűkebb körben használja az egyes szabályokat. Ha például egy ismert vírusminta a **sendmail** alkalmazáshoz kapcsolódik, akkor a szabály célportjának a **sendmail** SMTP 25-ös portját adja meg. Ez lehetővé teszi a többi hálózati forgalom átengedését anélkül, hogy a mintaillesztés fölöslegesen foglalná a rendszer erőforrásait.

A **genfilt** parancs képes felismerni és értelmezni a ClamAV néhány verziójában használt mintaformátumot.

Kapcsolódó tájékoztatás:

genfilt parancs

mkfilt parancs

ClamAV webhely

Minták típusai:

A minták három alaptípusba sorolhatók: szöveges, hexadecimális és fájl. A rendszer a mintaillesztési szűrőszabályokat csak a bejövő csomagokra alkalmazza.

Szövegminta

A szöveges szűrőminták az alábbihoz hasonló ASCII karakterláncok.

```
GET /../../../../../../../../
```

Hexadecimális minta

Egy hexadecimális minta az alábbihoz hasonlóan néz ki:

```
0x33c0b805e0cd16b807e0cd1650558becc7460200f05d0733ffb8c800b9fffff3abb00150
e670e47132c0e67158fec03c8075f033c033c9b002fa99cd26fb4183f90575f5c3
```

Megjegyzés: A hexadecimális mintát a szöveges mintától az első két karaktere (0x) különbözteti meg.

Szövegmintákat tartalmazó fájlok

A fájl soronként egy szöveges vagy hexadecimális mintát tartalmaz. A http://www.clamav.net címről letölthető néhány példafájl.

Megelőző port és hoszt szűrőszabályok:

Megelőző szűrőszabály beállításával megakadályozhatja, hogy egy távoli hoszt vagy a távoli hoszt - port pár hozzáférhessen a helyi géphez.

A megelőző szűrőszabály dinamikusan létrehoz egy aktív szabályt, amely egy adott feltétel teljesülése esetén megtagadja a távoli hoszt (vagy hoszt/pár kombináció) számára a helyi gép elérését.

A támadásokat gyakran előzi meg portkeresés, ezért az ilyen viselkedés felismerésével a megelőző port szűrőszabályok különösen hatékonyak lehetnek a támadások elhárításában.

Ha például a hely hoszt nem használja az időszerverhez tartozó 37-es portot, akkor a távoli hosztok nem próbálják meg elérni ezt a portot, kivéve portkeresés esetén. Helyezzen el egy megelőző szűrőszabályt a 37-es portra, így ha egy távoli hoszt megkísérli elérni ezt a portot, akkor a megelőző szűrő létrehoz egy aktív szűrőt, amely a szabály **expiration time** mezőjében megadott ideig blokkolja a hosztról érkező további kéréseket.

Ha egy megelőző szabály **expiration time** mezőjének értéke 0, akkor a dinamikusan létrehozott hatályos megelőző szabály nem jár le.

Megjegyzés:

1. A megelőző port szűrőszabály által megadott lejárati idő csak a létrehozott aktív szabályra érvényes.
2. A dinamikusan létrehozott szabályok az **lsfilt -a** paranccsal jeleníthetők meg.

Megelőző hoszt szűrőszabályok

Ha egy megelőző hoszt szűrőszabály feltételei teljesülnek, akkor a dinamikusan létrehozott aktív szabály a megadott lejárati ideig blokkolja az adott hosztról érkező hálózati forgalmat.

Megelőző portszűrőszabályok

Ha egy megelőző port szűrőszabály feltételei teljesülnek, akkor a dinamikusan létrehozott tényleges szabály a megadott lejárati ideig blokkolja a távoli hoszt adott portjáról érkező hálózati forgalmat.

Állapot-nyilvántartó szűrő szabályok:

Az állapot-nyilvántartó szűrők megvizsgálják a hálózati csomagok forrás és cél címét, állapotát és a portszámokat. Az IF, ELSE és ENDIF szűrőszabályok alkalmazásával a csomag fejléce alapján nem csak az adott csomaggal, hanem a teljes munkamenettel kapcsolatban képesek szűrési döntéseket hozni.

Az állapot-nyilvántartó ellenőrzés a bejövő és kimenő csomagokat egyaránt vizsgálja. Az állapot-nyilvántartó szűrőszabályok **mkfilt -u** paranccsal elvégzett aktiválása után az ELSE blokkok szabályai kerülnek kiértékelésre

egészen addig, amíg egyszer teljesül az IF szabály. Ha az IF szabály vagy feltétel egyszer teljesült, akkor a rendszer ettől kezdve az IF blokk szabályait használja mindaddig, amíg az adminisztrátor újra nem aktiválja a szabályt az **mkfilt -u** paranccsal.

A **ckfilt** parancs ellenőrzi az állapot-nyilvántartó szűrőszabály szintaxisát, és megjeleníti a szerkezetét az alábbihoz hasonló szemléletes formában:

```
%ckfilt -v4
Beginning of IPv4 filter rules.
2. Szabály
IF 3. Szabály
    IF 4. Szabály
        5. Szabály
    ELSE 6. Szabály
        7. Szabály
    ENDIF 8. Szabály
ELSE 9. Szabály
    10. Szabály
ENDIF 11. Szabály
0. Szabály
```

Időzített szabályok:

Az időzített szabályok megadják, hogy az aktív szabályt hány másodperccel azután kell alkalmazni, hogy az az **mkfilt -v [4|6] -u** paranccsal létrejött.

A lejárati időt a **genfilt -e** paranccsal lehet megadni. További információkat az **mkfilt** és **genfilt** parancs leírásában talál.

Megjegyzés: Az időzítőknek nincs hatásuk az IF, ELSE és ENDIF szabályokon. Ha a megelőző szabályban lejárati időt ad meg, akkor az csak az általa kezdeményezett aktív szabályra érvényes. A megelőző szabályoknak nincs lejárati idejük.

Szűrőszabályok elérése SMIT eszközből

A szabályokat beállíthatja a SMIT eszközből.

A szűrőszabályok SMIT eszközből használatához tegye a következőket:

1. A parancssorba írja be az alábbi parancsot:smitty ipsec4
2. Válassza ki a **Fejlett IP biztonság beállítása** menüpontot.
3. Válassza ki a **IP biztonsági szűrő szabályok beállítása** menüpontot.
4. Válassza ki a **IP biztonsági szűrőszabályok hozzáadása** menüpontot.

```
                        Add an IP Security Filter Rule

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                 [Entry Fields]
* Rule Action                                        [permit]                +
* IP Source Address                                  []
* IP Source Mask                                     []
  IP Destination Address                             []
  IP Destination Mask                                []
* Apply to Source Routing? (PERMIT/inbound only)     [yes]                   +
* Protocol                                           [all]                   +
* Source Port / ICMP Type Operation                  [any]                   +
* Source Port Number / ICMP Type                     [0]                      #
* Destination Port / ICMP Code Operation             [any]                   +
* Destination Port Number / ICMP Type                [0]                      #
* Routing                                            [both]                  +
* Direction                                          [both]                  +
* Log Control                                        [no]                    +
* Fragmentation Control                              [0]                     +
* Interface                                          []                      +
  Expiration Time (sec)                              []                       #
  Pattern Type                                       [none]                  +
  Pattern / Pattern File                             []
  Description                                        []


Where "Pattern Type" may be one of the following
  x   none                                                                   x#
  x   pattern                                                                x
  x   file                                                                   x
  x   Anti-Virus patterns
```

Az action mező lehetséges értékei: permit, deny, shun_host, shun_port, if, else, endif.

Ha minta fájlt van megadva, akkor annak olvashatónak kell lennie a szűrőszabályok aktiválásakor (**mkfilt -a** parancs). A szűrőszabályok az /etc/security/ipsec_filter adatbázisban tárolódnak.

AIX biztonsági szakértő

Az AIX Security Expert a biztonsági beállítások központja (TCP, NET, IPSEC, rendszer és megfigyelés).

Az AIX Security Expert egy rendszerbiztonság fokozó eszköz. Ez a **bos.aixpert** fájlkészlet része. Az AIX Security Expert egyszerű menübeállításokat biztosít a Magas szintű biztonság, az Közepes szintű biztonság, az Alacsony szintű biztonság és az AIX szabvány beállítások biztonsághoz. A menük több mint 300 biztonsági konfigurációs beállítást integrálnak, de továbbra is lehetővé teszik minden egyes biztonsági elem kezelését a képzett adminisztrátorok számára. Az AIX Security Expert a sok biztonságnöveléssel foglalkozó dokumentáció elolvasása és az egyes biztonsági elemek külön-külön való kezelése nélkül teszi lehetővé a megfelelő szintű biztonság megvalósítását.

Az AIX Security Expert segítségével pillanatképet készíthet a biztonsági konfigurációról. A pillanatkép felhasználásával azonos biztonsági konfigurációt állíthat be más rendszereken. Így idő takarít meg, és biztosíthatja, hogy minden rendszeren megfelelő biztonsági konfiguráció legyen a vállalati környezetben.

Az AIX Security Expert a SMIT-ből vagy az **aixpert** paranccsal futtatható.

AIX Security Expert beállítások

Az alábbi alapszintű biztonsági beállítások használhatók:

Magas szintű biztonság
: Magas szintű biztonság

Közepes szintű biztonság
Közepes szintű biztonság

Alacsony szintű biztonság
Alacsony szintű biztonság

Speciális biztonság
A felhasználó által megadott egyéni biztonság

AIX szabvány beállítások
Eredeti alapértelmezett rendszerbiztonság

Biztonság visszavonása
Egyes AIX Security Expert konfigurációs beállításokat nem lehet visszavonni

Biztonság ellenőrzése
Részletes jelentést ad az aktuális biztonsági beállításokról

AIX biztonsági szakértő biztonság fokozása

A biztonság fokozása a biztonság erősítésével vagy magasabb szintű biztonság alkalmazásával védi a rendszer összes elemét.

A biztonság fokozása segít biztosítani, hogy minden biztonsági konfigurációs döntés és beállítás helyénvaló és megfelelő legyen. Az AIX rendszer biztonságának fokozásához több száz biztonsági beállítás módosítására lehet szükség.

Az AIX Security Expert egy menüvel központosítja a hatékony, általános biztonsági konfigurációs beállításokat. Ezek a beállítások megfelelően védett UNIX rendszerek átfogó vizsgálatán alapulnak. Számos biztonsági környezethez alapértelmezett biztonsági szintek (Magas szintű biztonság, Közepes szintű biztonság és Alacsony szintű biztonság) választhatók, de a képzett adminisztrátorok minden egyes biztonsági konfigurációs beállítást külön is kezelhetnek.

Túl magas biztonsági szint beállítása egy rendszeren letilthatja a szükséges szolgáltatásokat. A **telnet** és **rlogin** parancs például a Magas biztonsági szinten le van tiltva, mivel a bejelentkezési jelszót titkosítás nélkül továbbítja a hálózaton. Ha a rendszer alacsony biztonsági szintre van beállítva, akkor a rendszer sebezhetővé válhat. Mivel minden vállalatnak saját, egyéni biztonsági követelményei vannak, ezért a Magas, Közepes és Alacsony szintű biztonság inkább egy jó kiindulópont, mint az adott vállalat biztonsági követelményeinek pontos megfelelője.

Az AIX Security Expert használatának gyakorlati megközelítése az éles környezethez hasonló tesztrendszer (realisztikus tesztkörnyezetben) létrehozása, amelyben a termék alkalmazásra kerül. Telepítse a szükséges üzleti alkalmazásokat és futtassa az AIX Security Expertt a grafikus felhasználói felületen keresztül. Az AIX Security Expert ezt a rendszer megbízható állapotban való futtatásával elemzi. A választott biztonsági beállítástól függően az AIX Security Expert engedélyezi a portvégignézés elleni védelmet, bekapcsolja a megfigyelést, blokkolja az üzleti alkalmazások vagy más szolgáltatások által nem használt hálózati portokat más biztonsági beállítással együtt. Miután újra tesztelte a rendszert ezekkel a biztonsági konfigurációkkal, a rendszer készen áll az éles környezetben való telepítésére. Ezen felül a rendszer biztonsági irányelvét és konfigurációját megadó AIX Security Expert XMLfile segítségével egyszerűen megvalósítható pontosan ugyanaz a konfiguráció a környezet hasonló rendszerein.

A biztonság fokozásáról az a következő kiadványban talál további információkat: NIST Special Publication 800-70, NIST Security Configurations Checklist Program for IT Products.

Alapértelmezésben védett

Az Alapértelmezésben védett (SbD) lehetőség a minimális szoftverhalmaz telepítésének alapelve biztonságos konfigurációban.

Az AIX alapértelmezésben védett (SbD) telepítési lehetőség a TCP kliens és szerver fájlkészletek egyszerűsített változatát telepíti, amely nem tartalmaz támadható parancsokat és fájlokat. A **bos.net.tcp.client** és **bos.net.tcp.server** fájlkészlet az SbD telepítés része és tartalmazza az összes parancsot illetve fájlt azon alkalmazások kivételével,

amelyek lehetővé teszik a jelszavak hálózaton keresztüli átvitelét sima szöveges formájában, mint például a **telnet** és **ftp**. Ezen felül a használható alkalmazások, mint például az **rsh**, **rcp** és **sendmail**, ki vannak zárva az SbD fájlkészletekből.

Az SbD telepítés végső automatikus folyamata az AIX Security Expert magas szintű biztonsági konfiguráció beállítások megadása. Ez az **aixpert** parancs /etc/firstboot parancsfájlból történő futtatásával hajtható végre: /usr/sbin/aixpert -f /etc/security/aixpert/core/SbD.xml -p 2>/etc/security/aixpert/log/firstboot.log

A gép átállítható SbD módból az ODM *SbD_STATE* változójának sbd_disable értékre állításával, a **bos.net.tcp.client** és **bos.net.tcp.server** fájlkészlet újbóli telepítésével, valamint a rendszer AIX Security Expert segítségével történő alapértelemzett biztonsági szintre állításával.

Az átállítási és megőrzési telepítés segítségével nem érhető el SbD telepített rendszer. Az SbD egy különálló telepítési menüútvonal.

Megjegyzés: Amikor egy SbD módban lévő rendszer frissít egy szervicsomaggal, a rendszer a frissítést követően már nem lesz SbD módban.

Az SbD telepítési lehetőség nélkül beállítható egy biztonságosan beállított rendszer. Az AIX Security Expert magas, közepes vagy alacsony szintű biztonsági beállítások például a szabályos telepítés során beállíthatók.

Az SbD telepítésű rendszer és a AIX Security Expert magas szintű biztonsági konfigurációval rendelkező szabályos telepítés közötti különbséget legjobban a **telnet** parancs vizsgálata mutatja. A **telnet** parancs mindkét esetben tiltott. SbD telepítésben a **telnet** bináris fájl vagy alkalmazás nincs telepítve a rendszeren.

SbD telepítés esetén a következő szolgáltatások a telepítés idején vagy nem kerülnek telepítésre a rendszeren, vagy le vannak tiltva. Mivel a szolgáltatások egy része nincs telepítve a rendszeren, ezek a parancsok nem érhetők el és nem futtathatók a rendszeren. Ha ezekre a parancsokra és programokra szükség van, akkor ne használja az SbD telepítési lehetőséget. Ezen felül ha bármely parancsfájlnak, távoli programnak vagy függő fájlkészletnek szüksége van ezekre a programokra és parancsokra, akkor ne használja az SbD telepítési lehetőséget.

Szolgáltatás	**Program**	**Argumentumok**
bootps	/usr/sbin/bootpd	bootpd /etc/bootp
comsat	/usr/sbin/comsat	comsat
exec	/usr/sbin/rexecd	rexecd
finger	/usr/sbin/fingerd	fingerd
ftp	/usr/sbin/ftpd	ftpd
instsrv	/u/netinst/bin/instsrv	instsrv -r /tmp/netinstalllog /u/netinst/scripts
login	/usr/sbin/rlogind	rlogind
netstat	/usr/bin/netstat	netstat -f inet
ntalk	/usr/sbin/talkd	talkd
pcnfsd	/usr/sbin/rpc.pcnfsd	pcnfsd
rexd	/usr/sbin/rpc.rexd	rexd
rquotad	/usr/sbin/rpc.rquotad	rquotad
rstatd	/usr/sbin/rpc.rstatd	rstatd
rusersd	/usr/lib/netsvc/rusers/rpc.rusersd	rusersd
rwalld	/usr/lib/netsvc/rwall/rpc.rwalld	rwalld
shell	/usr/sbin/rshd	rshd
sprayd	/usr/lib/netsvc/spray/rpc.sprayd	sprayd

Szolgáltatás	Program	Argumentumok
systat	/usr/bin/ps	ps -ef
talk	/usr/sbin/talkd	talkd
telnet	/usr/sbin/telnetd	telnetd -a
tftp	/usr/sbin/tftpd	tftpd -n
uucp	/usr/sbin/uucpd	uucpd

Létezik néhány funkció az IBM Systems Director Console for AIX programban, beleértve a HealthMetrics portletet is, amelyek nem érhetők el az AIX operációs rendszer SbD módú futtatása során. Ezeket a funkciókat azon fájlkészletek telepítésével engedélyezheti, melyek a funkció futtatásához szükségesek.

Biztonsági irányelv terjesztése LDAP protokollon keresztül

Az LDAP segítségével terjeszthetők az AIX Security Expert XML konfigurációs fájlok. Az AIX Security Expert segítségével a biztonsági konfigurációt egyik rendszerről a másikra másolhatja. Ez lehetővé teszi, hogy a hasonló rendszerek azonos biztonsági konfigurációval rendelkezzenek. Ez a konzisztencia csökkenti a biztonság gyengeségeit.

A javasolt gyakorlat a következő: az AIX Security Expert segítségével konfiguráljon egy rendszert és állítsa be a biztonsági irányelvet a vállalati biztonsági irányelveknek megfelelően, amelyben a rendszer működik. A konfiguráció az /etc/security/aixpert/core/appliedaixpert.xml fájlban kerül lementésre. A fájl ezután áthelyezhető egy beállított és megbízható LDAP szerverre. Ehhez az LDAP szerverhez csatlakozással rendelkező más rendszerek automatikusan feltérképezik ezt az XML konfigurációs fájlt az **aixpertldap** parancson keresztül.

A meglévő LDAP szerver frissíthető az aixpert sémával az aixpert konfigurációs XML fájlok csatlakoztatott kliensek felé történő terjesztése érdekében. Ha az LDAP szerver nem rendelkezik frissített aixpert sémával, akkor frissítse az aixpert sémát az LDAP-ben a következő paranccsal: ldapmodify -c -D <bindDN> -w <bindPwd> -i /etc/security/ldap/sec.ldif Miután az LDAP szerver frissítve lett az aixpert sémával, a kliensek az XML konfigurációs fájlokat LDAP-n az **aixpertldap** parancs -u paraméterével helyezhetik el. Ezeket a konfigurációs fájlokat saját kezűleg kell frissíteni.

Megjegyzés: Ez a szolgáltatás arra épül, hogy a megbízhatósági modell LDAP helyben megtalálható. Az LDAP írására jogosult felhasználók módosíthatják a különböző gépek felhasználói által feltöltött adatokat. Ehhez hasonlóan ha egy LDAP kliens biztonságilag sebezhető, akkor ez felhasználható más LDAP kliensek biztonsági állapotának beolvasásához és megismeréséhez a klienshez tartozó AIX Security Expert XML konfigurációs fájlok olvasásával.

Az appliedaixpert.xml fájl például elmenthetők az LDAP szerveren **BranchOfficeSecurityProfile** néven. Vagy egy másképp beállított appliedaixpert.xml fájl elmenthető **InternetDirectAttachedSystemsProfile** néven. Mivel más LDAP kapcsolattal rendelkező rendszerek az AIX Security Expertvel lettek beállítva, ezek a biztonsági profilok automatikusan megjelenítésre kerülnek menüpontokként. Ez lehetővé teszi, hogy a rendszeradminisztrátor kiválassza azt a biztonsági profilt, ami legjobban megfelel a környezetben a vállalati biztonsági irányelvek irányelvein belül.

Majd az AIX Security Expert segítségével biztonságossá tehető a rendszer. A rendszeren megvalósított biztonsági konfigurációs beállítások teljes listája az /etc/security/aixpert/core/appliedaixpert.xml fájlban található. Ez a fájl a rendszer biztonsági irányelve. A biztonsági irányelv az AIX Security Expert biztonság ellenőrzése beállításának használatakor kerül összehasonlításra. Ez a biztonsági irányelv más rendszerekre is átmásolható és más rendszereken is alkalmazható, ezáltal az IT környezet rendszereinek biztonsága konzisztenssé tehető. A biztonsági irányelv kétféleképp másolható át más rendszerekre: kézzel vagy az LDAP protokollon keresztül.

AIX biztonsági szakértő biztonsági irányelv másolása

Az AIX Security Expert segítségével a biztonsági irányelvet egyik rendszerről a másikra másolhatja.

Az AIX Security Expertt futtathatja egy rendszeren, majd ugyanazt a biztonsági irányelvet alkalmazhatja más rendszereken is. Tegyük fel, hogy Bob hat AIX rendszeren szeretné alkalmazni az AIX Security Expertt. Alkalmazza a biztonsági beállításokat az egyik rendszeren (Alpha) Magas, Közepes, Alacsony, Speciális vagy AIX szabvány

beállítások biztonsággal. Teszteli a rendszer kompatibilitását a környezetben. Ha elégedett a beállításokkal, akkor ugyanezeket a beállítások név alapján más AIX rendszereken is alkalmazhatja. Átmásolja a beállításokat az Alpha rendszerről arra a rendszerre, amelyen ugyanolyan biztonsági beállításokat szeretne alkalmazni. Ehhez átmásolja az /etc/security/aixpert/core/appliedaixpert.xml fájlt az Alpha rendszerről a másik rendszerre.

Megjegyzés: Ne másolja a fájlt a másik rendszer azonos könyvtárába ugyanazon a néven, mert az **aixpert** parancs felülírja az /etc/security/aixpert/core/appliedaixpert.xml fájlt, mivel az valósítja meg a biztonsági irányelvet.

Ehelyett másolja az Alpha biztonsági irányelvét az /etc/security/aixpert/custom/ könyvtárba. Ez lehetővé teszi, hogy más rendszer megtekintse és alkalmazza az Alpha biztonsági irányelvét az AIX Security Expert rendszerfelügyeleti grafikus felhasználói felületen keresztül, vagy közvetlenül az **aixpert** paranccsal.

Ha például az Alpha appliedaixpert.xml biztonsági irányelve más rendszerekre /etc/security/aixpert/custom/*AlphaPolicy* néven került, akkor az aixpert -f /etc/security/aixpert/custom/*AlphaPolicy* parancs azonnal alkalmazza ezt a biztonsági irányelvet és a rendszer ugyanazzal a biztonsági konfigurációval fog rendelkezni, mint az Alpha. Ezen felül ha az Alpha biztonsági irányelve ebben a könyvtárban található, akkor az látható és a többi rendszer rendszerkezelési konzolján keresztül, az Aix biztonsági szakértő -> Áttekintés és feladatok -> Egyéni beállítások -> AlphaPolicy útvonallal alkalmazható.

Személyre szabható biztonsági irányelv felhasználó által megadott AIX biztonsági szakértő XML szabályokkal

Az XML fájlok segítségével egyedi biztonsági irányelvek állíthatók be.

Az AIX Security Expert dinamikusan felismeri ezeket az XML fájlokat. A létrehozott egyéni XMLsecurity irányelvfájlokat az /etc/security/aixpert/custom/ könyvtárba kell helyezni egy leíró fájllal együtt. Ezáltal az AIX Security Expert konzol grafikus felületen keresztüli elérésekor a grafikus XML szolgáltatások gazdag készlete az aixpert DTD-ben teljes egészében elfogadásra kerül.

A DTD a következő:

```
<?xml version='1.0'?>

<!--START-->

<!ELEMENT AIXPertSecurityHardening (AIXPertEntry+)>

<!-- Az AIXPertEntry a következő elemek egy példányát tartalmazhatja csak. -->

<!ELEMENT AIXPertEntry (AIXPertRuleType,
  AIXPertDescription, AIXPertPrereqList, AIXPertCommand,
  AIXPertArgs,AIXPertGroup)>

<!-- Az AIXPertEntry nevének egyedinek kell lennie. -->

<!ATTLIST AIXPertEntry
 name ID #REQUIRED
 function CDATA ""
>

<!ELEMENT AIXPertRuleType EMPTY>
<!ATTLIST AIXPertRuleType    type (LLS|MLS|HLS|DLS|SCBPS|Prereq) "DLS">
<!ELEMENT AIXPertDescription (#PCDATA)>
<!ELEMENT AIXPertPrereqList (#PCDATA)>
<!ELEMENT AIXPertCommand (#PCDATA)>
<!ELEMENT AIXPertArgs (#PCDATA)*>
<!ELEMENT AIXPertGroup (#PCDATA)*>
```

Az AIXPertEntry név egyedi név az XMLfile-on belül. Ez a név a kiválasztható grafikus gomb neve a fájl rendszerkonzolon keresztüli, Aix biztonsági szakértő -> Áttekintés és feladatok -> Egyéni beállítások -> <xml file=""></xml> útvonallal való elérésekor.

<!ELEMENT AIXPertRuleType EMPTY>
Ezt az XML fájlt egyéniként kell megadni.

<!ATTLIST AIXPertRuleType type (LLS|MLS|HLS|DLS|SCBPS|Prereq|Custom) "DLS"
Ezt az XML fájlt egyéniként kell megadni.

<!ELEMENT AIXPertDescription (#PCDATA)>
A fenti grafikus felületen keresztüli megjelenítéskor a leírás szövege előugró ablakként jelenik meg, és az egeret a rendszer a gombra helyezi.

<!ELEMENT AIXPertPrereqList (#PCDATA)>
Kiválasztható egy előfeltétel szabály ehhez a szabályhoz. Az előfeltétel szabálynak 0-t kell visszaadni, mielőtt az aixpert megvalósítja ezt a szabályt. Ha az XML fájlt grafikus felületen keresztül jeleníti meg, akkor ez a szabály kiszürkített lesz, ha az előfeltétel szabály nincs kielégítve. Ha előfeltétel-szabályt hoz létre, akkor az AIXPertRuleType elemnek 'Prereq' értékkel kell rendelkeznie.

Az előfeltétel szabály AIXPertDescription mezejének azt kell leírnia, hogy mit kell tenni az előfeltétel szabály kielégítéséhez. Ha az Egyéni szabályok kiszürkítve jelennek meg, mivel az előfeltétel szabályok egyike nem teljesült, akkor a rendszer a felhasználó számára megjeleníti az előfeltétel szabály leírás előugró ablakát, amely megmutatja, hogy a felhasználónak mit kell tenni az előfeltétel kijavítása érdekében.

<!ELEMENT AIXPertCommand (#PCDATA)>
Az elemnek az aixpert által a biztonsági szabályhoz végrehajtandó parancsot és annak teljes elérési útját kell tartalmaznia, például: /usr/bin/ls.

<!ELEMENT AIXPertArgs (#PCDATA)*>
Az elemnek argumentumokat kell tartalmazni a fenti parancshoz, például: -l

<!ELEMENT AIXPertGroup (#PCDATA)*>
Az aixpert szabályok grafikus felületeken keresztüli megjelenítés esetén csoportosíthatók. A szabályok általános halmaza például a "Hálózati biztonság" AIXPertGroup nevet adja meg.

Gyenge jelszavak szigorú ellenőrzése

Az AIX szolgáltatás a jelszavak módosításakor ellenőrzi a jelszavak gyengeségét. Ha ez a lehetőség az AIX Security Expert rendszerhez ki lett választva, akkor további jelszóellenőrzés kerül végrehajtásra, amikor a felhasználó kiválasztja vagy módosítja a jelszót. Ez az ellenőrzés az angol szótár szavai és a US összeírás alapján kiválasztott 1000 legáltalánosabb US keresztnév alapján történik.

Az AIX biztonsági szakértő által támogatott COBIT vezérlési célok

Az AIX Security Expert támogatja a SOB-COBIT követendő eljárás biztonsági szintet a magas, közepes, alacsony, alapértelmezett AIX és a speciális biztonsági beállításokon felül.

A United States Congress törvénybe iktatta a '2002-es Sarbanes-Oxley törvényt' a befektetők védelme érdekében a vállalatok által kiadott pénzügyi információk megbízhatóságának és pontosságának javításával. A COBIT vezérlési célok szolgáltatás segítséget nyújt a rendszeradminisztrátorok számára az IT rendszerek beállításában, karbantartásában és megfigyelésében, a törvénynek való megfelelés érdekében. A SOX Konfigurációsegéd az aixpert parancssoron keresztül érhető el. Ez a szolgáltatás a Sarbanes-Oxley törvény SOX 404-es szakaszának betartásában segít, de az AIX biztonsági szakértő SOX konfigurációsegédje automatikusan megvalósítja a SOX 404-es, belső vezérlések szakasz COBIT követendő eljárásaihoz általánosan hozzárendelt biztonsági beállításokat. Ezen felül az AIX biztonsági szakértő SOX megfigyelési szolgáltatást biztosít, amely jelenti a megfigyelőnek, hogy a rendszer megfelelően van-e ily módon beállítva. A szolgáltatás lehetővé teszi a rendszerkonfiguráció automatizálását, amely elősegíti az IT SOX-nak való megfelelést és a megfigyelési folyamat automatizálását.

Mivel a SOX nem nyújt segítséget azzal kapcsolatban, hogy az IT-nek hogyan kell megfelelnie a 404-es szakasznak, az IT iparág a www.isaca.org/ által részletezett meglévő irányításra koncentrál. Pontosabban a Control Objectives for Information and related Technology (COBIT) által leírt IT irányításra.

Az AIX Security Expert a következő vezérlési célokat támogatja:

- Jelszóirányelv betartatása
- Sértés- és biztonsági tevékenységjelentések
- Rosszindulatú és jogosulatlan szoftver megakadályozása, felismerése és kijavítása
- Tűzfal architektúra és kapcsolatok nyilvános hálózatokkal

Az AIX Security Expert nem támogatja az egyes vezérlési célok alatt megadott attribútumok mindegyikét. A támogatott attribútumokat és a megfelelő vezérlési célokat a következő táblázatok foglalják össze:

Jelszóirányelv betartatása

Leírás	Biztonsági beállítás
Maximális jelszóéletkor	maxage=13
Jelszóelőzmény foganatosítása	histsize=20
Minimális jelszóéletkor	minage=1
Minimális jelszóhossz	minlen=8
Legalább 6 karaktert tartalmaz	Minalpha=6
Hasonlóság a régi jelszóhoz	mindiff=4
Jelszólejárati figyelmeztetés napjai	pwdwarntime=14

Biztonsági sértések és tevékenységjelentés

Leírás	Biztonsági beállítás	Megjegyzések
Engedélyezett megfigyelés	igen	
Nincsenek közvetlen root bejelentkezések	igen	
Megfigyelés engedélyezése a privilegizált kiterjesztéshez	igen	Az AIXpert kihasználja a USER_SU megfigyelési eseményt. Győződjön meg róla, hogy az esemény be van kapcsolva.

Rosszindulatú szoftver felismerése és kijavítása

Az AIX Security Expert kihasználja az AIX megbízható szoftvervégrehajtási szolgáltatást annak ellenőrzése érdekében, hogy a szoftvert más módosította-e. A **trustchk** parancs ellenőrzi a Megbízható szoftveradatbázisban bejegyzett objektumok konzisztenciáját.

Tűzfalbeállítás

Az AIX Security Expert bekapcsolja az IPSec protokollt és engedélyezi a szűrőszabályokat a portleolvasások elkerülése érdekében. A lezárt portokat a következő lista tartalmazza:

Szolgáltatás	Leírás
Tcp/11, udp/11	Systat
Tcp/13, udp/13	Daytime
(RFC 867) Tcp/19, udp/19	Karakterelőállító
Tcp/25	Simple Mail Transfer (SMTP)
Tcp/43, udp/43	Who Is (becenév)
Tcp/63, udp/63	Whois++
Tcp/67, udp/67	Rendszerbetöltési protokoll szerver (bootps)
Tcp/68, udp/68	Rendszerbetöltési protokoll kliens (bootpc)

Szolgáltatás	Leírás
Tcp/69, udp/69	Triviális fájlátvitel
(tftp) Tcp/79, udp/79	Finger
Tcp/87	Saját terminál hivatkozás
Tcp/110	Postahivatal protokoll – 3-as változat (POP3)
Udp/111	SUN Távoli eljáráshívás
Tcp/113	Hitelesítési szolgáltatás (auth)
Udp/123	Hálózati idő protokoll
Udp/161	SNMP
Udp/162	SNMPTRAP
Tcp/194	Internet csevegés protokoll
Tcp/443	TLS-en/SSL-en keresztüli http protokoll
Tcp/511	PassGo
Tcp/514	Cmd (parancsértelmező)
Tcp/520	Kiterjesztett fájlnévszerver (efs)
Tcp/540	Uucpd (uucp)
Tcp/546	DHCPv6 kliens
Tcp/547	DHCPv6 szerver
Tcp/555	Dsf
tcp/559	TEEDTAP
tcp/593	HTTP RPC Ep Map
udp/635	RLS Dbase
tcp/666	Mdqs
tcp/777	Multiling HTTP
tcp/901	SNMPNSMERES
tcp/902	IDEAFARM-CHAT
tcp/903	IDEAFARM-CATCH
tcp/1024	Fenntartott

COBIT vezérlési célok alkalmazása AIX biztonsági szakértő segítségével

Az **aixpert –l s** parancs kiadásával alkalmazhatja az SCBPS szintet a rendszerre. Ennek nyomkövetési naplója az **AIXpert_apply** esemény bekapcsolásával állítható elő. A hibák (az előfeltétel- és alkalmazási hiba egyaránt) jelentésre kerülnek az **stderr** kimeneten és a nyomkövetési alrendszeren, amennyiben engedélyezett.

SOX-COBIT-megfelelési ellenőrzés, megfigyelés és előzetes megfigyelési szolgáltatás

Az **aixpert –c –l s** paranccsal ellenőrizheti a rendszer SOX-COBIT-megfelelését. Az AIX Security Expert csak a támogatott vezérlési céloknak való megfelelést ellenőrzik. Az ellenőrzés során talált sértések jelentésre kerülnek. Alapértelmezésben a sértések az **stderr** kimenetre kerülnek.

Ugyanazzal a paranccsal (**aixpert –c –l s**) ellőállíthatja a SOX-COBIT-megfelelés megfigyelési jelentést. Megfigyelési jelentés előállításához állítsa be és engedélyezze a megfigyelési alrendszert. Győződjön meg róla, hogy az **AIXpert_check** megfigyelési esemény be van kapcsolva. A megfigyelési alrendszer beállítása után futtassa újra az

aixpert –c –l s parancsot. A parancs előállítja a megfigyelési naplót minden meghiúsult vezérlési célhoz. A nyomkövetési napló **Állapot** mezője mint **meghiúsult** lesz megjelölve. A napló a hiba okát is tartalmazza, amely az **auditpr** parancs **-v** paramétere segítségével jeleníthető meg.

A **–p** paraméter **aixpert –c –l s** parancshoz adása a sikeres vezérlési célokat is tartalmazza a megfigyelési jelentésben. Ezeknek a naplóbejegyzéseknek az állapotmezője **Ok** értéket tartalmaz.

Az **aixpert -c -l s -p** parancs segítségével előállítható egy részletes SOX-COBIT-megfelelési megfigyelési jelentés.

Attól függetlenül, hogy a **–p** paraméter meg van-e adva, létrejön egy összegzés rekord. Az összegzés rekord a feldolgozott szabályok számával, a meghiúsult szabályok számával (a talált nem megfelelések példányai), és a biztonsági szinttel kapcsolatos információkat tartalmaz, amelyre vonatkozóan a rendszer ellenőrzésre kerül (ebben a példányban ez az SCBPS).

AIX biztonsági szakértő jelszó házirend szabályok csoport

Az AIX Security Expert bizonyos beállításokat biztosít a jelszó házirendhez.

Az erős jelszó házirend a rendszerbiztonság egyik alapköve. A jelszó házirend biztosítja hogy a jelszavakat nehezen lehessen kitalálni (a jelszavak alfanumreikus karakterekből, számokból és speciális karakterkből álljanak), rendszeresen lejárjanak, és hogy a lejárat után ne lehessen őket újra felhasználni. Az alábbi táblázat az egyes biztonsági beállítások jelszó házirend szabályait mutatja be.

20. táblázat: AIX Security Expert jelszó házirend szabályai

Művelet gomb neve	Meghatározás	AIX Security Expert által beállított érték	Visszavonás
Karakterek minimális száma	Az /etc/security/user fájl **mindiff** attribútumát állítja be. Ez az attribútum határozza meg, hogy az új jelszavakban minimum hány olyan karaktert kell használni, amelyek nem szerepeltek a régi jelszóban.	**Magas szintű biztonság** 4 **Közepes szintű biztonság** 3 **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs korlát	Igen
Jelszó minimális kora	Az /etc/security/user fájl **minage** attribútumát állítja be. Ez az attribútum határozza meg, hogy minimum hány hétnek kell eltelnie a jelszó lejártához.	**Magas szintű biztonság** 1 **Közepes szintű biztonság** 4 **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs korlát	Igen
Jelszó maximális kora	Az /etc/security/user fájl **maxage** attribútumát állítja be. Ez az attribútum határozza meg, hogy maximum hány hétnek kell eltelnie a jelszó lejártához.	**Magas szintű biztonság** 13 **Közepes szintű biztonság** 13 **Alacsony szintű biztonság** 52 **AIX szabvány beállítások** Nincs korlát	Igen

20. táblázat: AIX Security Expert jelszó házirend szabályai (Folytatás)

Művelet gomb neve	Meghatározás	AIX Security Expert által beállított érték	Visszavonás
Jelszó minimális hossza	Az /etc/security/user fájl **minlen** attribútumát állítja be. Ez az attribútum határozza meg a jelszó minimális hosszát.	**Magas szintű biztonság** 8 **Közepes szintű biztonság** 8 **Alacsony szintű biztonság** 8 **AIX szabvány beállítások** Nincs korlát	Igen
Alfabetikus karakterek minimális száma	Az /etc/security/user fájl **minalpha** attribútumát állítja be. Ez az attribútum határozza meg az alfanumerikus karakterek minimális számát a jelszóban.	**Magas szintű biztonság** 2 **Közepes szintű biztonság** 2 **Alacsony szintű biztonság** 2 **AIX szabvány beállítások** Nincs korlát	Igen
Jelszó alaphelyzetbe állítási ideje	Az /etc/security/user fájl **histexpire** attribútumát állítja be. Ez az attribútum határozza meg, hogy hány hétig lehet a jelszót alaphelyzetbe állítani.	**Magas szintű biztonság** 13 **Közepes szintű biztonság** 13 **Alacsony szintű biztonság** 26 **AIX szabvány beállítások** Nincs korlát	Igen
Adott karakter maximális számú előfordulása a jelszóban	Az /etc/security/user fájl **maxrepeats** attribútumát állítja be. Ez az attribútum határozza meg, hogy az egyes karakterek maximum hányszor szerepelhetnek a jelszóban.	**Magas szintű biztonság** 2 **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** 8	Igen
Jelszó újrafelhasználási ideje	Az /etc/security/user fájl **histsize** attribútumát állítja be. Ez az attribútum határozza meg, hogy a felhasználó hány előzőleg használt jelszót nem használhat újra.	**Magas szintű biztonság** 20 **Közepes szintű biztonság** 4 **Alacsony szintű biztonság** 4 **AIX szabvány beállítások** Nincs korlát	Igen

20. táblázat: AIX Security Expert jelszó házirend szabályai (Folytatás)

Művelet gomb neve	Meghatározás	AIX Security Expert által beállított érték	Visszavonás
Jelszócserére rendelkezésre álló idő a lejárat után	Az /etc/security/user fájl **maxexpired** attribútumát állítja be. Ez az attribútum határozza meg, hogy a **maxage** paraméterben beállított időtartam után a felhasználó hány hétig cserélheti le a lejárt jelszót.	**Magas szintű biztonság** 2 **Közepes szintű biztonság** 4 **Alacsony szintű biztonság** 8 **AIX szabvány beállítások** -1	Igen
Nem alfabetikus karakterek minimális száma	Az /etc/security/user fájl **minother** attribútumát állítja be. Ez az attribútum határozza meg a nem alfabetikus karakterek minimális számát a jelszóban.	**Magas szintű biztonság** 2 **Közepes szintű biztonság** 2 **Alacsony szintű biztonság** 2 **AIX szabvány beállítások** Nincs korlát	Igen
Jelszólejárat figyelmeztetési ideje	Az /etc/security/user fájl **pwdwarntime** attribútumát állítja be. Ez az attribútum határozza meg, hogy a rendszer hány nap után küld figyelmeztetést, hogy a jelszót le kell cserélni.	**Magas szintű biztonság** 5 **Közepes szintű biztonság** 14 **Alacsony szintű biztonság** 5 **AIX szabvány beállítások** Nincs korlát	Igen

AIX biztonsági szakértő felhasználói csoport rendszer és jelszó meghatározások csoport

Az AIX Security Expert bizonyos műveleteket hajt végre a felhasználói, csoport és jelszó meghatározásokon.

21. táblázat: AIX Security Expert felhasználói csoport rendszer és jelszó meghatározások

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
Csoport meghatározások ellenőrzése	Ellenőrzi a csoport meghatározások helyességét. A következő parancs futtatásával javítja és jelenti a hibákat: `% grpck -y ALL`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Igen **AIX szabvány beállítások** Nincs hatással rá	Nem

21. táblázat: AIX Security Expert felhasználói csoport rendszer és jelszó meghatározások (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
TCB frissítés	A **tcbck** paranccsal ellenőrzi és frissíti a TCB-t. A következő parancsot futtatja: `% tcbck -y ALL` **Megjegyzés:** Ha a TCB szükséges a rendszeren, akkor ez a szabály meghiúsul, ha a TCB nincs engedélyezve. Az előfeltétel szabály (prereqtcb) is meghiúsul egy figyelmeztetéssel. **Előfeltétel**: A rendszer telepítésekor a TCB-t ki kell választani.	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Igen **AIX szabvány beállítások** Igen	Nem

Fájl meghatározások ellenőrzése	A **sysck** paranccsal ellenőrzi a javítást és az /etc/objrepos/inventory fájlalapját:
	`% sysck -i -f \`
	`/etc/security/sysck.cfg.rte`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Igen
		Alacsony szintű biztonság
		Igen
		AIX szabvány beállítások
		Nincs hatással rá
			Nem

Jelszó meghatározások ellenőrzése	Ellenőrzi a jelszó meghatározások helyességét. A következő parancs futtatásával javítja és jelenti a hibákat: `% pwdck -y ALL`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Igen **AIX szabvány beállítások** Nincs hatással rá	Nem
Felhasználói meghatározások ellenőrzése	Ellenőrzi a felhasználói meghatározások helyességét. A következő parancs futtatásával javítja és jelenti a hibákat: `% usrck -y ALL`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Igen **AIX szabvány beállítások** Nincs hatással rá	Nem

AIX biztonsági szakértő bejelentkezési házirend ajánlások csoport

Az AIX Security Expert bizonyos beállításokat biztosít a bejelentkezési házirendhez.

Megjegyzés: A biztonsággal kapcsolatos tevékenységek felelősségre vonhatóságának biztosítása érdekében ajánlott hogy a felhasználók először a saját szokásos felhasználói azonosítójukkal jelentkezzenek be, és utána a **su** *parancs* segítségével futtassák root felhasználóként a parancsokat, és ne jelentkezzenek be root felhasználóként. A rendszer így társítani tudja a root fiók használatával futtatott tevékenységeket a különböző felhasználókhoz, ha több felhasználó is ismeri és használja a root jelszót.

22. táblázat: AIX Security Expert bejelentkezési házirend ajánlások

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
Sikertelen bejelentkezések közötti időtartam	Az /etc/security/login.cfg fájl **logininterval** attribútumát állítja be. Ez az attribútum határozza meg, hogy hány másodpercnek kell eltelnie egy porton a sikertelen bejelentkezések között, mielőtt a rendszer letiltaná a portot. Ha például a **logininterval** értéke 60 a **logindisable** értéke pedig 4, akkor a fiókot a rendszer abban az esetben zárolja, ha négy sikertelen bejelentkezés történik egy percen belül.	**Magas szintű biztonság** 300 **Közepes szintű biztonság** 60 **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs korlát	Igen
Bejelentkezési kísérletek száma a fiók zárolása előtt	Az /etc/security/user fájl **loginretries** attribútumának értékét állítja be. Ez az attribútum határozza meg, hogy a rendszer hány egymás utáni sikertelen bejelentkezés után zárolja a fiókot. A root felhasználóhoz ne állítsa be.	**Magas szintű biztonság** 3 **Közepes szintű biztonság** 4 **Alacsony szintű biztonság** 5 **AIX szabvány beállítások** Nincs korlát	Igen
Távoli root bejelentkezés	Az /etc/security/user fájl **rlogin** attribútumának értékét módosítja. Ez az attribútum határozza meg, hogy a root fiók bejelentkezhet-e távolról a rendszerre.	**Magas szintű biztonság** Hamis **Közepes szintű biztonság** Hamis **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igaz	Igen
Bejelentkezés újraengedélyezése zárolás után	Az /etc/security/login.cfg fájl **loginreenable** attribútumát állítja be. Ez az attribútum határozza mag, hogy a rendszer hány másodperc elteltével oldja fel a port zárolását, miután a portot a **logindisable** letiltotta.	**Magas szintű biztonság** 360 **Közepes szintű biztonság** 30 **Low Level Security** Nincs hatással rá **AIX szabvány beállítások** Nincs korlát	Igen
Bejelentkezés letiltása sikertelen bejelentkezési kísérletek után	Az /etc/security/login.cfg fájl **logindisable** attribútumát állítja be. Ez az attribútum határozza meg, hogy a rendszer hány sikertelen bejelentkezési kísérlet után zárolja a portot.	**Magas szintű biztonság** 10 **Közepes szintű biztonság** 10 **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs korlát	Igen

22. táblázat: AIX Security Expert bejelentkezési házirend ajánlások (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
Bejelentkezési időkorlát	Az /etc/security/login.cfg fájl **logintimeout** attribútumát állítja be. Ez az attribútum határozza meg, hogy mennyi idő áll rendelkezésre a jelszó begépelésére.	**Magas szintű biztonság** 30 **Közepes szintű biztonság** 60 **Alacsony szintű biztonság** 60 **AIX szabvány beállítások** 60	Igen
Sikertelen bejelentkezések közötti késleltetés	Az /etc/security/login.cfg fájl **logindelay** attribútumát állítja be. Ez az attribútum határozza meg a sikertelen bejelentkezések közötti késleltetést (másodpercekben). A rendszer további késleltetést alkalmaz minden egyes sikertelen bejelentkezés után. Ha például a **logindelay** paraméter értéke 5, akkor a terminál az első sikertelen bejelentkezés után 5 másodpercet várakozik a következő kérésig. A második sikertelen bejelentkezés után a terminál 10 másodpercet (2*5), a harmadik sikertelen bejelentkezés után pedig már 15 másodpercet (3*5) várakozik.	**Magas szintű biztonság** 10 **Közepes szintű biztonság** 4 **Alacsony szintű biztonság** 5 **AIX szabvány beállítások** Nincs korlát	Igen
Helyi bejelentkezés	Az /etc/security/user fájl **login** attribútumának értékét módosítja. Ez az attribútum határozza meg, hogy a root fiók bejelentkezhet-e konzolról a rendszerre.	**Magas szintű biztonság** Hamis **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igaz	Igen

AIX biztonsági szakértő megfigyelési házirend ajánlások csoport

Az AIX Security Expert speciális megfigyelési házirend beállításokat biztosít.

A többi biztonsági beállításhoz hasonlóan a bináris megfigyeléshez szükséges, hogy az elemzés (előfeltétel) szabályok megfeleljenek, még a Magas, Közepes vagy Alacsony biztonsági szint megfigyelési szabályainak alkalmazása előtt. A bináris megfigyeléshez az alábbi elemzési szabályoknak kell teljesülniük:

1. A megfigyelés előfeltétel szabályának ellenőriznie kell, hogy a megfigyelés éppen nem fut-e. Ha a megfigyelés már fut, akkor a megfigyelés már be van állítva, és a AIX Security Expertnek nem szabad módosítania a meglévő megfigyelési konfigurációt és eljárást.
2. A kötetcsoportban legalább 100 megabyte szabad helynek kell lennie, amely automatikusan bekapcsolásra kerül, vagy léteznie kell a /audit fájlrendszernek szintén legalább 100 megabyte-os méretben.

Ha a fenti előfeltételek teljesülnek és a megfigyelési beállítások az AIX Security Expert rendszerben kerültek kiválasztásra, akkor az AIX Security Expert beállítja és engedélyezi a megfigyelést a rendszeren a következő módon. Az AIX Security Expert **Bináris megfigyelés engedélyezése** művelete beállítja a megfigyelési házirendet. A megfigyelésnek engedélyezve kell lennie a rendszeren.

1. A /audit JFS fájlrendszert létre kell hozni és fel kell építeni a megfigyelés indítása előtt. A fájlrendszernek legalább 100 megabyte-osnak kell lennie.
2. A megfigyelést bináris módban kell futtatni. Az /etc/security/audit/config fájlt a következőképpen kell beállítani:

```
start:
            binmode = on
            streammode = off
bin:
```

```
        trail = /audit/trail
        bin1 = /audit/bin1
        bin2 = /audit/bin2
        binsize = 10240
        cmds = /etc/security/audit/bincmds
.
.
etc
```

3. Adja hozzá a megfigyelési bejegyzéseket a root és a szokásos felhasználókhoz a Magas, Közepes és az Alacsony szintű biztonsághoz.
4. A megfigyelést engedélyezni kell, ha Magas, Közepes vagy Alacsony szintű biztonsággal szeretné újraindítani a rendszert.
5. A létrehozott új felhasználóknál engedélyezni kell a megfigyelést a Magas, Közepes és Alacsony szintű biztonsághoz. Ehhez hozzá kell adni egy auditclasses bejegyzést az /usr/lib/security/mkuser.default fájl felhasználói szakaszához.
6. Az /audit fájlrendszer megtelésének megakadályozása érdekében hozzá kell adni egy **cronjob** bejegyzést.

A megfigyelés visszavonási szabályának le kell állítania a megfigyelést és le kell tiltania a megfigyelést az újraindításkor.

Az alábbi táblázat az AIX Security Expert által a **Bináris megfigyelés engedélyezése** beállításhoz megadott értékeket tartalmazza:

23. táblázat: AIX Security Expert által a Bináris megfigyelés engedélyezéséhez beállított értékek

Magas szintű biztonság	Közepes szintű biztonság	Alacsony szintű biztonság	AIX szabvány beállítások

Adja hozzá az alábbi megfigyelési bejegyzéseket a root és a szokásos felhasználókhoz:
`Root:`
`General`
`Src`
`Mail`
`Cron`
`Tcpip`
`Ipsec`
`Lvm`
`User:`
`General`
`Src`
`Cron`
`Tcpip`
 	
Adja hozzá a következő bejegyzést az /usr/lib/security/mkuser.default fájl felhasználói szakaszához, és így engedélyezze a megfigyelést az újonnan létrehozott felhasználóknál:
`auditclasses=general,SRC,\`
`cron,tcpip`
	Adja hozzá az alábbi megfigyelési bejegyzéseket a root és a szokásos felhasználókhoz:
	`Root:`
	`General`
	`Src`
	`Tcpip`
	`User:`
	`General`
	`Tcpip`
	
	Adja hozzá a következő bejegyzést az /usr/lib/security/mkuser.default fájl felhasználói szakaszához, és így engedélyezze a megfigyelést az újonnan létrehozott felhasználóknál:
	`auditclasses=general,`
	`tcpip`
		Adja hozzá az alábbi megfigyelési bejegyzéseket a root és a szokásos felhasználókhoz:
		`Root:`
		`General`
		`Tcpip`
		`User:`
		`General`
		
		Adja hozzá a következő bejegyzést az /usr/lib/security/mkuser.default fájl felhasználói szakaszához, és így engedélyezze a megfigyelést az újonnan létrehozott felhasználóknál:
		`auditclasses=general`
			Az /etc/security/audit/config fájl az alábbi bejegyzést tartalmazza:
			`default=login`
			
			A megfigyelési osztály bejelentkezés a következőképpen van megadva:
			`login = USER_SU,`
			`USER_Login,`
			`USER_Logout,`
			`TERM_Logout,`
			`USER_Exit`
			
			Megjegyzés: A szokásos beállítások letiltja a megfigyelést.

23. táblázat: AIX Security Expert által a Bináris megfigyelés engedélyezéséhez beállított értékek (Folytatás)

Magas szintű biztonság	Közepes szintű biztonság	Alacsony szintű biztonság	AIX szabvány beállítások
Adja hozzá az alábbi megfigyelési bejegyzéseket a root és a szokásos felhasználókhoz: **root:** general src mail cron tcpip ipsec lvm aixpert *Felhasználó:* general src cron tcpip Adja hozzá a következő bejegyzést az /usr/lib/security/mkuser.default fájl felhasználói szakaszához, és így engedélyezze a megfigyelést az újonnan létrehozott felhasználóknál: `auditclasses=general,SRC, cron,tcpip`	Adja hozzá az alábbi megfigyelési bejegyzéseket a root és a szokásos felhasználókhoz: **root:** general src tcpip aixpert *Felhasználó:* general tcpip Adja hozzá a következő bejegyzést az /usr/lib/security/mkuser.default fájl felhasználói szakaszához, és így engedélyezze a megfigyelést az újonnan létrehozott felhasználóknál: `auditclasses=general, tcpip`	Adja hozzá az alábbi megfigyelési bejegyzéseket a root és a szokásos felhasználókhoz: **root:** general tcpip aixpert *Felhasználó:* general Adja hozzá a következő bejegyzést az /usr/lib/security/mkuser.default fájl felhasználói szakaszához, és így engedélyezze a megfigyelést az újonnan létrehozott felhasználóknál: auditclasses=general	Igen

A cronjob jobnak minden órában le kell futnia, és ellenőriznie kell az /audit méretét. Ha a Szabad terület megfigyelése igaz értékre van állítva, akkor végre kell hajtani a Másolási műveletek nyomkövetési megfigyelését. A Szabad terület megfigyelésével kell ellenőrizni, hogy az /audit fájlrendszer nem telt-e meg. Ha az /audit fájlrendszer megtelt, akkor a rendszer elvégzi a Másolási műveletek nyomkövetési megfigyelését (letiltja a megfigyelést, biztonsági mentést készít az /audit/trail könyvtárról az /audit/trailEgySzinttelVissza könyvtárba, és ismét engedélyezi a nyomkövetést).

AIX biztonsági szakértő /etc/inittab bejegyzések csoport

Az AIX Security Expert néhány bejegyzést megjegyzéssé alakít az /etc/inittab fájlban, így ezek nem indulnak el a rendszerbetöltéskor.

24. táblázat: AIX Security Expert /etc/inittab bejegyzései

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
qdaemon engedélyezése / **qdaemon** letiltása	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inittab fájlban: `qdaemon:2:wait:/usr/bin/startsrc –sqdaemon`	**Magas szintű biztonság** Megjegyzéssé alakítja **Közepes szintű biztonság** Megjegyzéssé alakítja **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nem megjegyzéssé alakítja	Igen

24. táblázat: AIX Security Expert /etc/inittab bejegyzései (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
lpd démon letiltása / **lpd** démon engedélyezése	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inittab fájlban: `lpd:2:once:/usr/bin/startsrc -s lpd`	**Magas szintű biztonság** Megjegyzéssé alakítja **Közepes szintű biztonság** Megjegyzéssé alakítja **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nem megjegyzéssé alakítja	Igen
CDE letiltása / CDE engedélyezése	Ha a rendszeren nincs LFT beállítva, akkor megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inittab fájlban: `dt:2:wait:/etc/rc.dt`	**Magas szintű biztonság** Megjegyzéssé alakítja **Közepes szintű biztonság** Megjegyzéssé alakítja **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nem megjegyzéssé alakítja	Igen
piobe démon letiltása / **piobe** démon engedélyezése	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inittab fájlban: `piobe:2:wait:/usr/lib/lpd/pio/etc/pioinit >/dev/null 2>&1`	**Magas szintű biztonság** Megjegyzéssé alakítja **Közepes szintű biztonság** Megjegyzéssé alakítja **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nem megjegyzéssé alakítja	Igen

AIX biztonsági szakértő /etc/rc.tcpip beállításai csoport

Az AIX Security Expert néhány bejegyzést megjegyzéssé alakít az /etc/rc.tcpip fájlban, így ezek nem indulnak el a rendszerbetöltéskor.

Az alábbi táblázat azokat a bejegyzéseket tartalmazza az /etc/rc.tcpip fájlban, amelyeket a rendszer megjegyzéssé alakít, és amelyek így nem indulnak el a rendszerbetöltéskor.

25. táblázat: AIX Security Expert /etc/rc.tcpip beállításai

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
Levelező kliens letiltása / Levelező kliens engedélyezése	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/lib/sendmail "$src_running"`	**Magas szintű biztonság** Megjegyzéssé alakítja **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nem megjegyzéssé alakítja	Igen
Útválasztó démon letiltása	Megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/routed "$src_running" -q`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
mrouted démon letiltása	Megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/mrouted "$src_running"`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
timed démon letiltása	Megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/timed`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Igen **AIX szabvány beállítások** Igen	Igen
rwhod démon letiltása	Megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/rwhod "$src_running"`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen

25. táblázat: AIX Security Expert /etc/rc.tcpip beállításai (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
print démon letiltása	Megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/lpd "$src_running"`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
SNMP démon letiltása / SNMP démon engedélyezése	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/snmpd "$src_running"`	**Magas szintű biztonság** Megjegyzéssé alakítja **Közepes szintű biztonság** Megjegyzéssé alakítja **Alacsony szintű biztonság** SNMP démon letiltása **AIX szabvány beállítások** Nem megjegyzéssé alakítja	Igen
DHCP ügynök leállítása	Megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/dhcprd "$src_running"`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
DHCP szerver leállítása	Megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/dhcpsd "$src_running"`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
autoconf6 leállítása	Megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/autoconf6 ""`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen

25. táblázat: AIX Security Expert /etc/rc.tcpip beállításai (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
DNS démon letiltása	Megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/named "$src_running"`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
gated démon letiltása	Megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/gated "$src_running"`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Igen **AIX szabvány beállítások** Igen	Igen
DHCP kliens leállítása	Megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/dhcpd "$src_running"`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
DPID2 démon letiltása	Megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/dpid2 "$src_running"`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
NTP démon letiltása	Megjegyzéssé alakítja a következő bejegyzést az /etc/rc.tcpip fájlban: `start /usr/sbin/xntpd "$src_running"`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen

AIX biztonsági szakértő /etc/inetd.conf beállításai csoport

Az AIX Security Expert bizonyos bejegyzéseket megjegyzéssé alakít az /etc/inetd.conf fájlban.

Az AIX alapértelmezett telepítése számos olyan hálózati szolgáltatást engedélyez, amelyek veszélyeztethetik a rendszer biztonságát. Az AIX Security Expert úgy tiltja le a szükségtelen nem biztonságos szolgáltatásokat, hogy a megfelelő

sorokat megjegyzéssé alakítja az /etc/inetd.conf fájlban. Az AIX szabványos beállításoknál a bejegyzések nincsenek megjegyzéssé alakítva. Az alábbi táblázat a /etc/inetd.conf fájl megjegyzéssé alakított és nem megjegyzéssé alakított bejegyzéseit tartalmazza.

26. táblázat: AIX Security Expert /etc/inetd.conf beállításai

Művelet gomb neve	**Leírás**	**AIX Security Expert által beállított érték**	**Visszavonás**
sprayd letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban: `sprayd sunrpc_udp udp wait root \ /usr/lib/netsvc/`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
UDP chargen szolgáltatás letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban: `chargen dgram udp wait root internal`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
telnet letiltása / telnet engedélyezése	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban: `telnet stream tcp6 nowait root \ /usr/sbin/telnetd telnetd`	**Magas szintű biztonság** Megjegyzéssé alakítja **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nem megjegyzéssé alakítja	Igen
UDP Echo szolgáltatás letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban: `echo dgram udp wait root internal`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
tftp letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban: `tftp dgram udp6 SRC nobody \ /usr/sbin/tftpd tftpd -n`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen

26. táblázat: AIX Security Expert /etc/inetd.conf beállításai (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás

krshd démon letiltása	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`kshell stream tcp nowait root \`
	`/usr/sbin/krshd krshd`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

rusersd letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`rusersd sunrpc_udp udp wait root \`
	`/usr/lib/netsvc/`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Igen
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

rexecd letiltása az /etc/inetd.conf fájlban / **rexecd** engedélyezése az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`exec stream tcp6 nowait root \`
	`/usr/sbin/rexecd rexecd`
		Magas szintű biztonság
		Megjegyzéssé alakítja
		Közepes szintű biztonság
		Megjegyzéssé alakítja
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Nem megjegyzéssé alakítja
			Igen

POP3D letiltása	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`pop3 stream tcp nowait root \`
	`/usr/sbin/pop3d pop3d`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

pcnfsd letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`pcnfsd sunrpc_udp udp wait root \`
	`/usr/sbin/rpc.pcnfsd pcnfsd`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

26. táblázat: AIX Security Expert /etc/inetd.conf beállításai (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás

bootpd letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`bootps dgram udp wait root \`
	`/usr/sbin/bootpd`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Igen
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

rwalld letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`rwalld sunrpc_udp udp wait root \`
	`/usr/lib/netsvc/`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Igen
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

UDP discard szolgáltatás letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`discard dgram udp wait root \`
	`internal`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

TCP daytime szolgáltatás letiltása az /etc/inetd.conf fájlban / TCP daytime szolgáltatás engedélyezése az /etc/inetd.conf fájlban	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`daytime stream tcp nowait root \`
	`internal`
		Magas szintű biztonság
		Megjegyzéssé alakítja
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Nem megjegyzéssé alakítja
			Igen

netstat letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`netstat stream tcp nowait nobody \`
	`/usr/bin/netstat`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Igen
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

26. táblázat: AIX Security Expert /etc/inetd.conf beállításai (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás

rshd démon engedélyezése / **rshd** démon letiltása	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`shell stream tcp6 nowait root \`
	`/usr/sbin/rshd rshd rshd`
		Magas szintű biztonság
		Megjegyzéssé alakítja
		Közepes szintű biztonság
		Megjegyzéssé alakítja
		Alacsony szintű biztonság
		Megjegyzéssé alakítja
		AIX szabvány beállítások
		Nem megjegyzéssé alakítja
			Igen

cmsd szolgáltatás letiltása az /etc/inetd.conf fájlban / **cmsd** szolgáltatás engedélyezése az /etc/inetd.conf fájlban	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`cmsd sunrpc_udp udp wait root \`
	`/usr/dt/bin/rpc.cms cmsd`
		Magas szintű biztonság
		Megjegyzéssé alakítja
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Nem megjegyzéssé alakítja
			Igen

ttdbserver szolgáltatás letiltása az /etc/inetd.conf fájlban / **ttdbserver** szolgáltatás engedélyezése az /etc/inetd.conf fájlban	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`ttdbserver sunrpc_tcp tcp wait \`
	`root /usr/dt/bin/`
		Magas szintű biztonság
		Megjegyzéssé alakítja
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Nem megjegyzéssé alakítja
			Igen

uucpd letiltása az /etc/inetd.conf fájlban / **uucpd** engedélyezése az /etc/inetd.conf fájlban	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`uucp stream tcp nowait root \`
	`/usr/sbin/uucpd uucpd`
		Magas szintű biztonság
		Megjegyzéssé alakítja
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Nem megjegyzéssé alakítja
			Igen

UDP time szolgáltatás letiltása az /etc/inetd.conf fájlban / UDP time szolgáltatás engedélyezése az /etc/inetd.conf fájlban	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban: `time dgram udp wait root internal`	**Magas szintű biztonság** Megjegyzéssé alakítja **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nem megjegyzéssé alakítja	Igen

26. táblázat: AIX Security Expert /etc/inetd.conf beállításai (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás

TCP time szolgáltatás letiltása az /etc/inetd.conf fájlban / TCP time szolgáltatás engedélyezése az /etc/inetd.conf fájlban	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`time stream tcp nowait root \`
	`internal`
		Magas szintű biztonság
		Megjegyzéssé alakítja
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Nem megjegyzéssé alakítja
			Igen

rexd letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`rexd sunrpc_tcp tcp wait root \`
	`/usr/sbin/tpc.rexd.rexd rexd`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Igen
		Alacsony szintű biztonság
		Igen
		AIX szabvány beállítások
		Igen
			Igen

TCP chargen szolgáltatás letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`chargen stream tcp nowait root \`
	`internal`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

rlogin letiltása az /etc/inetd.conf fájlban / **rlogin** engedélyezése az /etc/inetd.conf fájlban	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`login stream tcp6 nowait root \`
	`/usr/sbin/rlogind rlogind`
		Magas szintű biztonság
		Megjegyzéssé alakítja
		Közepes szintű biztonság
		Megjegyzéssé alakítja
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Nem megjegyzéssé alakítja
			Igen

talk letiltása az /etc/inetd.conf fájlban	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`talk dgram udp wait root \`
	`/usr/sbin/talkd talkd`
		Magas szintű biztonság
		Megjegyzéssé alakítja
		Közepes szintű biztonság
		Megjegyzéssé alakítja
		Alacsony szintű biztonság
		Megjegyzéssé alakítja
		AIX szabvány beállítások
		Nem megjegyzéssé alakítja
			Igen

26. táblázat: AIX Security Expert /etc/inetd.conf beállításai (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás

fingerd letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`finger stream tcp nowait nobody \`
	`/usr/sbin/fingerd fingerd`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Igen
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

FTP letiltása / FTP engedélyezése	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`ftp stream tcp6 nowait root \`
	`/usr/sbin/ftpd ftpd`
		Magas szintű biztonság
		Megjegyzéssé alakítja
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Nem megjegyzéssé alakítja
			Igen

IMAPD letiltása	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`imap2 stream tcp nowait root \`
	`/usr/sbin/imapd imapd`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

comsat letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`comsat dgram udp wait root \`
	`/usr/sbin/comsat comsat`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

rquotad letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`rquotad sunrpc_udp udp wait root \`
	`/usr/sbin/rpc.rquotad`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Igen
		Alacsony szintű biztonság
		Igen
		AIX szabvány beállítások
		Igen
			Igen

26. táblázat: AIX Security Expert /etc/inetd.conf beállításai (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
UDP daytime szolgáltatás letiltása az /etc/inetd.conf fájlban / UDP daytime szolgáltatás engedélyezése az /etc/inetd.conf fájlban	Megjegyzéssé vagy nem megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban: `daytime dgram udp wait root internal`	**Magas szintű biztonság** Megjegyzéssé alakítja **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nem megjegyzéssé alakítja	Igen

krlogind letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`klogin stream tcp nowait root \`
	`/usr/sbin/krlogind krlogind`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

TCP Discard szolgáltatás letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`discard stream tcp nowait root \`
	`internal`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Nincs hatással rá
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

TCP echo szolgáltatás letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban: `echo stream tcp nowait root internal`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen

sysstat letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban:
	`systat stream tcp nowait nodby \`
	`/usr/bin/ps ps -ef`
		Magas szintű biztonság
		Igen
		Közepes szintű biztonság
		Igen
		Alacsony szintű biztonság
		Nincs hatással rá
		AIX szabvány beállítások
		Igen
			Igen

26. táblázat: AIX Security Expert /etc/inetd.conf beállításai (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
rstatd letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban: `rstatd sunrpc_udp udp wait root \ /usr/sbin/rpc.rstatd rstatd`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
dtspc letiltása az /etc/inetd.conf fájlban	Megjegyzéssé alakítja a következő bejegyzést az /etc/inetd.conf fájlban: `dtspc stream tcp nowait root \ /usr/dt/bin/dtspcd`	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen

AIX biztonsági szakértő parancsok SUID-jének letiltása csoport

Az SUID bitkészlettel alapértelmezésben az alábbi parancsok kerülnek telepítésre. A Magas, Közepes és Alacsony biztonsági szintnél ez a bit nincs beállítva. Az AIX szabvány beállításoknál az SUID bit visszaállításra kerül ezeknél a parancsoknál.

27. táblázat: AIX Security Expert parancsok SUID-jének letiltása

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
hls_filepermgr	Fájljogosultság-kezelő: Az **fpm** parancsot magas szintű beállítással futtatja a setuid és setgid privilegizált parancsokból való eltávolítása érdekében	Magas szintű biztonság	Igen
mls_filepermgr	Fájljogosultság-kezelő: Az **fpm** parancsot közepes szintű beállítással futtatja a setuid és setgid privilegizált parancsokból való eltávolítása érdekében	Közepes szintű biztonság	Igen
lls_filepermgr	Fájljogosultság-kezelő: Az **fpm** parancsot alacsony szintű beállítással futtatja a setuid és setgid privilegizált parancsokból való eltávolítása érdekében	Alacsony szintű biztonság	Igen

AIX biztonsági szakértő távoli szolgáltatások letiltása csoport

Az AIX Security Expert letiltja a Magas és a Közepes szintű biztonság nem biztonságos parancsait.

Az alábbi parancsok és démonok gyakran használhatók biztonsági kibúvók keresésére. A Magas és Közepes szintű biztonságnál ezek a nem biztonságos parancsok végrehajtás tiltása engedélyeket kapnak, a démonok pedig le vannak tiltva. Az Alacsony szintű biztonság ezekre a parancsokra és démonokra nincs hatással. Az AIX szabvány beállításokban ezeknek a parancsoknak és démonoknak a használata engedélyezett.

- **rcp**
- **rlogin**
- **rsh**
- **tftp**
- **rlogind**
- **rshd**

- **tftpd**

28. táblázat: AIX Security Expert távoli szolgáltatások letiltása

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
Nem biztonságos démonok engedélyezése	Ha a TCB engedélyezve van, akkor végrehajtási engedélyeket állít be az **rlogind**, **rshd** és **tftpd** démonokra, és frissíti a sysck adatbázist ezeknek a démonoknak a mód bitjével. Ha a TCB nincs engedélyezve, akkor végrehajtás engedélyt állít be az **rlogind**, **rshd** és **tftpd** démonokon.	**Magas szintű biztonság** Nincs hatással rá **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs hatással rá	Igen
Nem biztonságos parancsok letiltása	1. Ha a TCP engedélyezve van, akkor eltávolítja az **rcp**, **rlogin**, **rsh** és **tftp** parancsok végrehajtási engedélyeit, és frissíti a **sysck** adatbázist ezeknek a parancsoknak a mód bitjével. Ha a TCB nincs engedélyezve, akkor eltávolítja az **rcp**, **rlogin** és **rsh** parancsok végrehajtás engedélyeit. 2. Leállítja az **rcp**, **rlogin**, **rsh**, **tftp** és **uftp** parancsok aktuális példányait, hacsak nem a parancsok egyike az AIX Security Expert szülőfolyamata. 3. Hozzáadja a **tcpip:** szakaszt az /etc/security/config fájlhoz, így korlátozza a **.netrc** használatát az **ftp** és **rexec** parancsokban.	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs hatással rá	Igen
Nem biztonságos parancsok engedélyezése	1. Ha a TCP engedélyezve van, akkor végrehajtási engedélyeket állít be az **rcp**, **rlogin**, **rsh** és **tftp** parancsokhoz, és frissíti a **sysck** adatbázist ezeknek a parancsoknak a mód bitjével. Ha a TCB nincs engedélyezve, akkor beállítja az **rcp**, **rlogin** és **rsh** parancsok végrehajtás engedélyeit. 2. Eltávolítja az /etc/security/config fájlt.	**Magas szintű biztonság** Nincs hatással rá **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
Nem biztonságos démonok letiltása	1. Ha a TCP engedélyezve van, akkor eltávolítja az **rlogind**, **rshd** és **tftpd** démonok végrehajtási engedélyeit, és frissíti a **sysck** adatbázist ezeknek a démonoknak a mód bit változásaival. Ha a TCB nincs engedélyezve, akkor végrehajtási engedélyeket állít be az **rlogind**, **rshd** és **tftpd** démonokra. 2. Leállítja az **rlogind**, **rshd** és **tftpd** démonok aktuális példányait, hacsak nem a démonok egyike az AIX Security Expert szülőfolyamata.	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs hatással rá	Igen
NFS démon leállítása	• Eltávolít minden NFS felépítést • Letiltja az NFS-t • Eltávolítja az NFS indító parancsfájlját az /etc/inittab helyről	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs hatással rá	Igen

28. táblázat: AIX Security Expert távoli szolgáltatások letiltása (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
NFS démon engedélyezése	• Exportálja az /etc/exports összes bejegyzését • Bejegyzést ad hozzá az /etc/inittab fájlhoz az /etc/rc.nfs futtatásához a rendszer újraindításakor • Azonnal futtatja az /etc/rc.nfs fájlt	**Magas szintű biztonság** Nincs hatással rá **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen

AIX biztonsági szakértő hitelesítést nem igénylő hozzáférés eltávolítása csoport

Az AIX támogat néhány olyan szolgáltatást, amelyek nem igényelnek felhasználói hitelesítést a hálózatra való bejelentkezéshez.

Az /etc/hosts.equiv fájl és a helyi **$HOME**/.rhosts fájlok határozzák meg, hogy mely hosztok és felhasználói fiókok futtathatnak jelszó nélkül távoli parancsokat a helyi rendszeren. Ha erre a képességre nincs kifejezetten szükség, akkor ezeknek a fájloknak a tartalmát törölni kell.

29. táblázat: AIX Security Expert hitelesítést nem igénylő hozzáférés eltávolítása

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
rhosts és netrc szolgáltatások eltávolítása	Az .rhosts és .netrc fájlok sima szövegként tárolják a felhasználói neveket és jelszavakat, ami kihasználható.	**Magas szintű biztonság** Távolítsa el az .rhosts és .netrc fájlokat az összes felhasználó saját könyvtárából, a root felhasználóéból is. **Közepes szintű biztonság** Távolítsa el az .rhosts és .netrc fájlokat az összes felhasználó saját könyvtárából, a root felhasználóéból is. **Alacsony szintű biztonság** Távolítsa el az .rhosts és .netrc fájlokat a root felhasználó saját könyvtárából. **AIX szabvány beállítások** Távolítsa el az .rhosts és .netrc fájlokat az összes felhasználó saját könyvtárából, a root felhasználóéból is.	Igen
Bejegyzések eltávolítása az /etc/hosts.equiv fájlból	Az /etc/hosts.equiv fájl a **$HOME**/.rhosts fájllal együtt határozza meg, hogy az idegen hosztok mely felhasználói futtathatnak távolról parancsokat a helyi rendszeren. Ha valaki az idegen hoszton megismeri a felhasználói nevet és a hosztnevet, akkor könnyen futtathat távoli parancsokat a helyi rendszeren hitelesítés nélkül.	**Magas szintű biztonság** Távolítsa el az összes bejegyzést az /etc/hosts.equiv fájlból. **Közepes szintű biztonság** Távolítsa el az összes bejegyzést az /etc/hosts.equiv fájlból. **Alacsony szintű biztonság** Távolítsa el az összes bejegyzést az /etc/hosts.equiv fájlból. **AIX szabvány beállítások** Távolítsa el az összes bejegyzést az /etc/hosts.equiv fájlból.	Igen

AIX biztonsági szakértő hálózati beállítások hangolása csoport

A hálózati beállítások megfelelő értékre állítása a biztonság fontos része. Egy hálózati attribútum 0 értékre állítja letiltja a beállítást, 1 értékre állítása pedig engedélyezi.

Az alábbi táblázat a Magas, Közepes és Alacsony szintű biztonság hálózati attribútum beállításait tartalmazza. A táblázat bemutatja azt is, hogy egy adott hálózati beállítás ajánlott értéke hogyan biztosíthatja a hálózat biztonságát.

30. táblázat: AIX Security Expert hálózati beállítások hangolása a hálózat biztonsága érdekében

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
ipsrcrouteforward hálózati beállítás	Megadja, hogy a rendszer továbbítja-e a forrás útválasztási csomagokat. Az ipsrcrouteforward letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.	**Magas szintű biztonság** 0 **Közepes szintű biztonság** 0 **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** 1	Igen
ipignoreredirects hálózati beállítás	Megadja, hogy a rendszer feldolgozza-e a megkapott átirányításokat.	**Magas szintű biztonság** 1 **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs korlát	Igen
clean_partial_conns hálózati beállítás	Megadja, hogy a rendszer elkerüli-e a szinkronizáló karakter (SYN) támadásokat.	**Magas szintű biztonság** 1 **Közepes szintű biztonság** 1 **Alacsony szintű biztonság** 1 **AIX szabvány beállítások** Nincs korlát	Igen
ipsrcrouterecv hálózati beállítás	Megadja, hogy a rendszer elfogadja-e a forrásból továbbított csomagokat. Az ipsrcrouterecv letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.	**Magas szintű biztonság** 0 **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs korlát	Igen
ipforwarding hálózati beállítás	Megadja, hogy a kernel továbbítsa-e a csomagokat. Az ipforwarding letiltása megakadályozza, hogy az átirányított csomagok távoli hálózatokba jussanak el.	**Magas szintű biztonság** 0 **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs korlát	Igen

30. táblázat: AIX Security Expert hálózati beállítások hangolása a hálózat biztonsága érdekében (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
ipsendredirects hálózati beállítás	Megadja, hogy a kernel küldjön-e átirányítási jelzéseket. Az ipsendredirects letiltása megakadályozza, hogy az átirányított csomagok távoli hálózatokba jussanak el.	**Magas szintű biztonság** 0 **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** 1	Igen
ip6srcrouteforward hálózati beállítás	Megadja, hogy a rendszer továbbítja-e a forrás útválasztási IPv6 csomagokat. Az ip6srcrouteforward letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.	**Magas szintű biztonság** 0 **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** 1	Igen
directed_broadcast hálózati beállítás	Megadja, hogy megengedett-e az átjáróhoz irányított üzenetszórás. A directed_broadcast letiltása segít megakadályozni, hogy az átirányított csomagok távoli hálózatokba jussanak el.	**Magas szintű biztonság** 0 **Közepes szintű biztonság** 0 **Alacsony szintű biztonság** 0 **AIX szabvány beállítások** Nincs korlát	Igen
tcp_pmtu_discover hálózati beállítás	Engedélyezi vagy letiltja az útvonal MTU feltérképezést a TCP alkalmazásokban. A tcp_pmtu_discover letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.	**Magas szintű biztonság** 0 **Közepes szintű biztonság** 0 **Alacsony szintű biztonság** 0 **AIX szabvány beállítások** 1	Igen
bcastping hálózati beállítás	Engedélyezi a választ az üzenetszórási címre küldött ICMP visszhang csomagokra. A bcastping letiltása megakadályozza a smurf támadásokat.	**Magas szintű biztonság** 0 **Közepes szintű biztonság** 0 **Alacsony szintű biztonság** 0 **AIX szabvány beállítások** Nincs korlát	Igen

30. táblázat: AIX Security Expert hálózati beállítások hangolása a hálózat biztonsága érdekében (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
icmpaddressmask hálózati beállítás	Megadja, hogy a rendszer válaszol-e az ICMP címmaszk kérésekre. Az icmpaddressmask letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.	**Magas szintű biztonság** 0 **Közepes szintű biztonság** 0 **Alacsony szintű biztonság** 0 **AIX szabvány beállítások** Nincs korlát	Igen
udp_pmtu_discover hálózati beállítás	Engedélyezi vagy letiltja az elérési út maximális átviteli egység (MTU) feltérképezést az UDP alkalmazásoknál. Az udp_pmtu_discover letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.	**Magas szintű biztonság** 0 **Közepes szintű biztonság** 0 **Alacsony szintű biztonság** 0 **AIX szabvány beállítások** 1	Igen
ipsrcroutesend hálózati beállítás	Megadja, hogy az alkalmazások küldhetnek-e forrás útválasztású csomagokat. Az ipsrcroutesend letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.	**Magas szintű biztonság** 0 **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** 1	Igen
nonlocsrcroute hálózati beállítás	Megadja az Internet protokollnak (IP), hogy szigorúan forrás útválasztású csomagok címezhetők-e a helyi hálózaton kívüli hosztokra is. A nonlocsrcroute letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.	**Magas szintű biztonság** 0 **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs korlát	Igen
tcp_tcpsecure hálózati beállítás	Védi a TCP kapcsolatokat támadásokkal szemben. Értékek: • 0 = nincs védelem • 1 = hamis SYN küldése létesített kapcsolatnak • 2 = hamis RST küldése létesített kapcsolatnak • 3 = adatok betöltése létesített TCP kapcsolatba • 5-7 = fenti gyengeségek kombinációja	**Magas szintű biztonság** 7 **Közepes szintű biztonság** 7 **Alacsony szintű biztonság** 5 **AIX szabvány beállítások** Nincs korlát	Igen

30. táblázat: AIX Security Expert hálózati beállítások hangolása a hálózat biztonsága érdekében (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
Sockthresh hálózati beállítás	Megadja a hálózati memória használati korlátját. Új socket kapcsolat nem megengedett, ha a beállítható sockthresh értéket túllépi. Megadja a socketek számára lefoglalható hálózati memória maximális mennyiségét.	**Magas szintű biztonság** 60 **Közepes szintű biztonság** 70 **Alacsony szintű biztonság** 85 **AIX szabvány beállítások** Nincs korlát	Igen

A következő hálózati beállítások inkább a hálózati teljesítménnyel kapcsolatosak mint a hálózati biztonsággal.

31. táblázat: AIX Security Expert hálózati beállítások hangolása a hálózat teljesítménye érdekében

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
rfc1323 hálózati beállítás	Az rfc1323 hangolás lehetővé teszi a TCP ablak méretezését.	**Magas szintű biztonság** 1 **Közepes szintű biztonság** 1 **Alacsony szintű biztonság** 1 **AIX szabvány beállítások** Nincs korlát	Igen
tcp_sendspace hálózati beállítás	A tcp_sendspace hangolás megadja, hogy a küldő alkalmazás mennyi adatot pufferelhet a kernelben, mielőtt a rendszer blokkolná az alkalmazás küldés hívását.	**Magas szintű biztonság** 262144 **Közepes szintű biztonság** 262144 **Alacsony szintű biztonság** 262144 **AIX szabvány beállítások** 16384	Igen
tcp_mssdflt hálózati beállítás	Távoli hálózatokkal való kommunikációban használt alapértelmezett maximális szegmensméret.	**Magas szintű biztonság** 1448 **Közepes szintű biztonság** 1448 **Alacsony szintű biztonság** 1448 **AIX szabvány beállítások** 1460	Igen
extendednetstats hálózati beállítás	Bővebb statisztikát tesz lehetővé a hálózati memória szolgáltatásokról.	**Magas szintű biztonság** 1 **Közepes szintű biztonság** 1 **Alacsony szintű biztonság** 1 **AIX szabvány beállítások** Nincs korlát	Igen

31. táblázat: AIX Security Expert hálózati beállítások hangolása a hálózat teljesítménye érdekében (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
tcp_recvspace hálózati beállítás	A tcp_recvspace hangolás meghatározza, hogy a fogadó rendszer hány byte adatot pufferelhet a kernelben a fogadó socket sorban.	**Magas szintű biztonság** 262144 **Közepes szintű biztonság** 262144 **Alacsony szintű biztonság** 262144 **AIX szabvány beállítások** 16384	Igen
sb_max hálózati beállítás	Az sb_max hangolás felső korlátot állít be egy adott socket sorbaállított socket puffereinek számára. Ez határozza meg, hogy a küldő sockethez vagy a fogadó sockethez sorbaállított pufferek mennyi pufferterületet használnak.	**Magas szintű biztonság** 1048576 **Közepes szintű biztonság** 1048576 **Alacsony szintű biztonság** 1048576 **AIX szabvány beállítások** 1048576	Igen

AIX biztonsági szakértő IPsec szűrőszabályok csoport

Az AIX Security Expert az alábbi IPsec szűrőket biztosítja.

32. táblázat: AIX Security Expert IPsec szűrőszabályai

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
Hoszt kikerülése 5 percre	Kikerüli vagy blokkolja a hoszt ismerten sérülékeny tcp és udp portjait öt percre. A host öt percig nem fogadja az ezekre a portokra címzett csomagokat.	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs hatással rá	Igen
Hoszt védése a portkereséssel szemben	Megvédi a hosztot a portkeresésekkel szemben. A portkeresést végző távoli hosztokat öt percig kikerüli vagy blokkolja. Az ilyen távoli hosztokról érkező csomagokat öt percig nem fogadja.	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs hatással rá	Igen

AIX biztonsági szakértő egyéb csoport

Az AIX Security Expert számos egyéb biztonsági beállítást biztosít a Magas, Közepes és Alacsony szintű biztonsághoz.

33. táblázat: AIX Security Expert egyéb csoport

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
Pont eltávolítása az elérési út gyökérből	A **$HOME**/.profile, **$HOME**/.kshrc, **$HOME**/.cshrc és **$HOME**/.login fájlokban pontokat (.) keres a PATH környezeti változóban, és eltávolítja azokat, ha léteznek. **Megjegyzés:** A pontok eltávolítása csak akkor valósul meg, ha a fájl bejegyzése a PATH környezeti változóval kezdődik, és pontokat (.) tartalmaz. A fájl nem változik, ha a PATH környezeti változó más változókat tartalmaz, vagy ha egy parancsfájlból meghívott programból visszakapott értékre van beállítva. A következő példa olyan útvonal, amely nem fog módosulni, ahol a *pathprog* egy útvonal karaktersorozatot visszaadó program: `PATH="$(pathprog)"` Ebben az útvonalban a pontok a *pathprog* változó tartalmának feloldása előtt eltávolításra kerülnek, így a visszaadott útvonalban lévő pontok nem lesznek eltávolítva.	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Igen **AIX szabvány beállítások** Igen	Igen
Rendszerhozzáférés korlátozása	Biztosítja, hogy csak a root felhasználó futtathasson **cron** jobokat a rendszeren.	**Magas szintű biztonság** Csak a root felhasználót szerepelteti a cron.allow fájlban, és eltávolítja a cron.deny fájlt. **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Eltávolítja a cron.allow fájlt és törli az összes bejegyzést a cron.deny fájlból.	Igen
Pont eltávolítása az /etc/environment fájlból	Eltávolítja a pontokat (.) a **PATH** környezeti változóból az /etc/environment fájlban.	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Igen **AIX szabvány beállítások** Igen	Igen
Pont eltávolítása a nem root elérési útból	Az összes nem root felhasználónál eltávolítja a pontokat (.) a **$HOME**/.profile, **$HOME**/.kshrc, **$HOME**/.cshrc és **$HOME**/.login fájlok **PATH** környezeti változójából. **Megjegyzés:** A pontok eltávolítása csak akkor valósul meg, ha a fájl bejegyzése a PATH környezeti változóval kezdődik, és pontokat (.) tartalmaz. A fájl nem változik, ha a PATH környezeti változó más változókat tartalmaz, vagy ha egy parancsfájlból meghívott programból visszakapott értékre van beállítva. A következő példa olyan útvonal, amely nem fog módosulni, ahol a *pathprog* egy útvonal karaktersorozatot visszaadó program: `PATH="$(pathprog)"` Ebben az útvonalban a pontok a *pathprog* változó tartalmának feloldása előtt eltávolításra kerülnek, így a visszaadott útvonalban lévő pontok nem lesznek eltávolítva.	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs hatással rá	Nem

33. táblázat: AIX Security Expert egyéb csoport (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
Root felhasználó hozzáadása az /etc/ftpusers fájlhoz	Hozzáadja a root felhasználó nevet az /etc/ftpusers fájlhoz, így letiltja a távoli root **ftp** szolgáltatást.	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen
Root felhasználó eltávolítása az /etc/ftpusers fájlból	Eltávolítja a root bejegyzést az /etc/ftpusers fájlból, és így engedélyezi a root **ftp** szolgáltatást.	**Magas szintű biztonság** Nincs hatással rá **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Igen	Igen

Bejelentkezési hirnök beállítása	Ellenőrzi az /etc/security/login.cfg fájl és biztosítja, hogy ne legyen hirnök érték beállítva. Ha használ hírnököt, akkor a hírnököt módosítani kell. A hírnököt csak akkor lehet módosítani, ha a rendszer területi beállítása **en_US** vagy egyéb angol területi beállítás. Ha ez a feltétel teljesül, akkor a hírnök attribútum értékét az /etc/security/login.cfg fájl alapértelmezett szakaszában a következőre állítja:
	`Unauthorized use of this \`
	`system is prohibited.\nlogin:`
	
	Megjegyzés: A biztonsági beállítás csak az új szekciókra fog vonatkozni. A biztonsági beállítás nincs hatással arra a szekcióra, amelyben a beállítás megadásra került.
		Magas szintű biztonság
		herald="Unauthorized use of this system is prohibited.\nlogin:"
		Közepes szintű biztonság
		herald="Unauthorized use of this system is prohibited.\nlogin:"
		Alacsony szintű biztonság
		herald="Unauthorized use of this system is prohibited.\nlogin:"
		AIX szabvány beállítások
		herald=
			Igen

Vendég fiók eltávolítása	Magas, Közepes és Alacsony biztonság esetén eltávolítja a vendég fiókot és a vendég adatait is a számítógépről. Az AIX szabvány beállításoknál egy vendég fiók kerül létrehozásra a rendszeren. **Megjegyzés:** A rendszergazdának kifejezetten be kell állítani egy jelszót ehhez a fiókhoz, mivel az AIX Security Expert nem kezeli az ilyen interaktív felhasználói feladatokat.	**Magas szintű biztonság** Eltávolítja a vendég fiókot és az adatokat **Közepes szintű biztonság** Eltávolítja a vendég fiókot és az adatokat **Alacsony szintű biztonság** Eltávolítja a vendég fiókot és az adatokat **AIX szabvány beállítások** Hozzáadja a vendég fiókot a számítógépen.	Igen

33. táblázat: AIX Security Expert egyéb csoport (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
Crontab engedélyek	Biztosítja, hogy a root felhasználó **crontab** jobjait a root felhasználó tulajdonolja és csak a root felhasználó írhatja.	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Igen **AIX szabvány beállítások** Nincs hatással rá	Igen
X-Server hozzáférés engedélyezése	Kötelezővé teszi a hitelesítést az X-Server hozzáféréshez.	**Magas szintű biztonság** Hitelesítés kötelező **Közepes szintű biztonság** Hitelesítés kötelező **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nem szükséges	Nem
Objektum létrehozási engedélyek	Az /etc/security/user fájl **umask** attribútumát állítja be. Ez az attribútum határozza meg az alapértelmezett objektum létrehozási engedélyeket.	**Magas szintű biztonság** 077 **Közepes szintű biztonság** 027 **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** 022	Igen
Törzsfájl méretének beállítása	Az /etc/security/limits fájl **core** attribútumát állítja be. Ez az attribútum határozza meg a törzsfájl méretét a root felhasználó számára. **Megjegyzés:** A biztonsági beállítás csak az új szekciókra fog vonatkozni. A biztonsági beállítás nincs hatással arra a szekcióra, amelyben a beállítás megadásra került.	**Magas szintű biztonság** 0 **Közepes szintű biztonság** 0 **Alacsony szintű biztonság** 0 **AIX szabvány beállítások** 2097151	Igen
SED szolgáltatás engedélyezése	Engedélyezi a Veremvégrehajtás letiltása szolgáltatást és futtatja a **sedmgr** parancsot a megadott fájlokon. **Megjegyzés:** A szabály életbe lépéséhez a rendszert újra kell indítani.	**Magas szintű biztonság** setidfiles **Közepes szintű biztonság** Nincs hatással rá **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs hatással rá	

33. táblázat: AIX Security Expert egyéb csoport (Folytatás)

Művelet gomb neve	Leírás	AIX Security Expert által beállított érték	Visszavonás
Root jelszó integritásának ellenőrzése	Biztosítja, hogy a root jelszó ne legyen gyenge. A root dictionlist attribútumának értéke /etc/security/aixpert/dictionary/English, így a **passwd** parancs segítségével biztosítható, hogy a beállított root jelszó ne legyen gyenge.	**Magas szintű biztonság** Igen **Közepes szintű biztonság** Igen **Alacsony szintű biztonság** Nincs hatással rá **AIX szabvány beállítások** Nincs hatással rá	Igen

AIX biztonsági szakértő biztonság visszavonása

Néhány AIX Security Expert biztonsági beállítást és szabályt vissza lehet vonni.

Az alábbi AIX Security Expert biztonsági beállításokat és szabályokat nem lehet visszavonni:

- Jelszó meghatározások ellenőrzése Magas, Közepes és Alacsony szintű biztonságnál
- Felhasználó meghatározások ellenőrzése Magas, Közepes és Alacsony szintű biztonságnál
- Csoport meghatározások ellenőrzése Magas, Közepes és Alacsony szintű biztonságnál
- TCB frissítés Magas, Közepes és Alacsony szintű biztonságnál
- X-Server hozzáférés engedélyezése Magas, Közepes és Alacsony szintű biztonságnál
- Pont eltávolítása a nem root elérési útból Magas szintű biztonságnál és AIX szabvány beállításoknál
- Vendég fiók eltávolítása Magas, Közepes és Alacsony szintű biztonságnál

AIX biztonsági szakértő biztonság ellenőrzése

Az AIX Security Expert képes jelentéseket készíteni az aktuális rendszer- és hálózati biztonsági beállításokról.

Ha az AIX Security Expert (aixpert parancs) segítségével beállította a rendszert, akkor a Biztonság ellenőrzése beállítással készíthet jelentést a különböző konfigurációs beállításokról. Ha valamelyik beállítás ezek közül az AIX Security Expertn kívül módosításra került, akkor az AIX Security Expert Biztonság ellenőrzése beállítása az eltéréseket az /etc/security/aixpert/check_report.txt fájlba naplózza.

Tegyük fel hogy az **talkd** démon letiltásra került az /etc/inetd.conf fájlban az Alacsony szintű biztonság alkalmazásakor. Ha a **talkd** démon később engedélyezésre kerül és futtatja a Biztonság ellenőrzését, akkor ez az információ a következőképpen kerül naplózásra a check_report.txt fájlba:

```
coninetdconf.ksh: Service talk using protocol udp
should be disabled, however it is enabled now.
```

Ha az alkalmazott biztonsági beállítások nem változtak, akkor a check_report.txt fájl üres lesz.

A Biztonság ellenőrzését rendszeres időközönként futtatni kell, és az eredményt vizsgálatával meg kell határozni, hogy az AIX Security Expert biztonsági beállításainak alkalmazása óta megváltoztak-e a beállítások. A Biztonság ellenőrzését a főbb rendszermódosításokkor, így a szoftverek telepítésekor és frissítésekor is futtatni kell.

Kapcsolódó tájékoztatás:

aixpert parancs

AIX biztonsági szakértő fájlok

Az AIX Security Expert különböző fájlokat hoz létre és használ.

/etc/security/aixpert/core/aixpertall.xml
: Az összes lehetséges biztonsági beállítás XML listáját tartalmazza.

/etc/security/aixpert/core/appliedaixpert.xml
Az alkalmazott biztonsági beállítások XML listáját tartalmazza.

/etc/security/aixpert/core/secaixpert.xml
A kijelölt biztonsági beállítások XML listáját tartalmazza az AIX Security Expert grafikus felhasználói felülettel végzett feldolgozáskor.

/etc/security/aixpert/log/aixpert.log
Az alkalmalzott biztonsági beállítások nyomkövetési naplóját tartalmazza. Az AIX Security Expert nem használja a syslog naplót. Az AIX Security Expert közvetlenül az /etc/security/aixpert/log/aixpert.log fájlba ír.

Megjegyzés: Az AIX Security Expert XML és naplófájlokat a következő engedélyekkel hozza létre:

/etc/security/aixpert/
drwx------

/etc/security/aixpert/core/
drwx------

/etc/security/aixpert/core/aixpertall.xml
r--------

/etc/security/aixpert/core/appliedaixpert.xml

/etc/security/aixpert/core/secaixpert.xml

/etc/security/aixpert/log
drwx------

/etc/security/aixpert/log/aixpert.log
-rw-------

/etc/security/aixpert/core/secundoaixpert.xml
rw-------

/etc/security/aixpert/check_report.txt
rw-------

AIX biztonsági szakértő Magas szintű biztonság példahelyzet

Az alábbi példahelyzet az AIX Security Expert Magas szintű biztonságát mutatja be.

A biztonsági szintek AIX Security Expert nézete részben a Nemzeti szabvány és technológiai intézetNational *Security Configuration Checklists Progarm for IT Pruducts - Guidance for CheckLists Users and Developers* (keressen rá a kiadvány nevére a NIS webhelyén: http://www.nist.gov/index.html) dokumentumából származik. A Magas, Közepes és Alacsony szintű biztonság mást és mást jelent a különböző felhasználók számára. Fontos megismerni a rendszer működési környezetét. Ha túl magas biztonsági szintet választ, akkor kizárhatja magát a saját számítógépéből. Ha túl alacsony biztonsági szintet választ, akkor a számítógépet sebezhetővé válhat támadások esetén.

Ez a példa olyan környezetre vonatkozik, amely magas szintű biztonságot igényel. Bob egy Internet szolgáltatónál helyezi el a rendszerét. A rendszer közvetlenül az Internetre fog csatlakozni, HTTP szerverként fog futni, érzékeny felhasználói adatokat fog tartalmazni, és Bobnak távolról kell adminisztrálnia. A rendszert először elszigetelt helyi hálózaton kell beállítani és tesztelni, és csak utána lehet online állapotba helyezni az Internet szolgáltatónál.

Ebben a környezetben a magas szintű biztonság a megfelelő, de Bobnak el kell érnie távolról a rendszert. A magas szintű biztonság nem engedélyezi a **telnet**, **rlogin**, **ftp** és az egyéb olyan parancsokat, amelyek titkosítatlan jelszavakat továbbítanak a hálózaton. Ezeket a jelszavakat bárki könnyen elfoghatja az Interneten. Bobnak egy olyan biztonságos módszerre van szüksége, amellyel távolról bejelentkezhet (például **openssh**-ra). Bobnak el kell olvasnia a teljes AIX Security Expert dokumentációt, és meg kell határoznia, hogy a környezetének van-e olyan egyedi jellemzője, amelyet a

magas szintű biztonság kizár. Ha van, akkor megszüntetheti az adott elem kijelölését a részletes magas szintű biztonság panel megjelenítésekor. Bobnak be kell állítania és el kell indítania a HTTP szervert és az egyéb olyan szolgáltatásokat, amelyeket a rendszerével biztosítani szeretne.

Ha Bob kiválasztja a Magas szintű biztonságot, akkor az AIX Security Expert felismeri hogy a futó szolgáltatásokra szükség van, és nem blokkolja a hozzáférést ezek portjaihoz. A többi porthoz való hozzáférés sérülékennyé teheti a rendszert, ezért ezeket a portokat a magas szintű biztonság letiltja. A konfiguráció tesztelése után Bob gépe készen áll arra, hogy online állapotba kerüljön az Interneten.

AIX biztonsági szakértő Közepes szintű biztonság példahelyzet

Az alábbi példahelyzet az AIX Security Expert Közepes szintű biztonságát mutatja be.

Alice-nak fokoznia kell a biztonságot egy olyan rendszeren, amely a vállalati tűzfal mögött lévő vállalati hálózatra lesz csatlakoztatva. A hálózat biztonságos és megfelelően van adminisztrálva. A rendszer nagyszámú felhasználó fogja használni. A felhasználóknak **telnet** és **ftp** hozzáférésre van szükségük a rendszerhez. Alice alkalmazni szeretné az általános biztonsági beállításokat, például a portkeresés elleni védelmet és a jelszó lejáratot, de a rendszert nyitva szeretné tartani a távoli hozzáférési módok számára. Ebben a példahelyzetben a Közepes biztonsági szint a megfelelő Alice rendszere számára.

AIX biztonsági szakértő Alacsony szintű biztonság példahelyzet

Az alábbi példahelyzet az AIX Security Expert Alacsony szintű biztonságát mutatja be.

Bruce már egy ideje a rendszer adminisztrátora. A rendszer egy elkülönített, biztonságos helyi hálózaton található. A rendszert sokféle felhasználó használja sokféle szolgáltatáshoz. Bruce emelni szeretné a biztonsági szintet a minimális szintről, de semmilyen formában nem szakíthatja meg a rendszerhez való hozzáférést. Az Alacsony szintű biztonság a megfelelő Bruce gépe számára.

Biztonsági ellenőrzőlista

A fejezet ellenőrzőlistája az újonnan telepített vagy már meglévő rendszerek biztonsági tevékenységeit tartalmazza.

Bár ez a lista nem egy teljes biztonsági ellenőrzőlista, alapként szolgálhat a saját környezet biztonsági ellenőrzőlistájának kialakításához.

- Új rendszer telepítésekor az AIX rendszert biztonságos adathordozóról telepítse. Telepítéskor hajtsa végre az alábbiakat:
  - Szerverekre ne telepítse az asztali szoftvereket, például a CDE-t, GNOME-ot vagy KDE-t.
  - Telepítse a szükséges biztonsági javításokat, illetve az esetleg ajánlott karbantartási és technológiai szint javításokat. A legfrissebb szervizhirdetmények, biztonsági tanácsadás és a javításokkal kapcsolatos információk itt találhatók: IBM System p eServer terméktámogatás javítások webhelye (http://www.ibm.com/support/fixcentral).
  - Mentse el a rendszert a kezdeti telepítés után és tegye a mentést biztonságos helyre.
- Dolgozza ki a korlátozott hozzáférésű fájlok és könyvtárak hozzáférés felügyeleti listáit.
- Tiltsa le a szükségtelen felhasználói és rendszerfiókokat (például daemon, bin, sys, adm, lp vagy uucp). A fiókok törlése nem ajánlott, mert ilyenkor törlődnek a fiókkal kapcsolatos információk is, például a felhasználói azonosítók és felhasználónevek, amelyekre például a rendszermentés adatai hivatkozhatnak. Ha létrehozunk egy felhasználót egy törölt felhasználói azonosítóval, akkor a mentés visszaállítása után az új felhasználó nem kívánt jogokhoz juthat a visszaállított rendszeren.
- Rendszeresen tekintse át az /etc/inetd.conf, /etc/inittab, /etc/rc.nfs és /etc/rc.tcpip fájlokat és töröljön belőlük minden szükségtelen démont és szolgáltatást.
- Ellenőrizze, hogy az alábbi fájlok jogosultságai helyesen vannak-e beállítva:

```
-rw-rw-r-- root     system  /etc/filesystems
-rw-rw-r-- root     system  /etc/hosts
-rw------- root     system  /etc/inittab
```

```
-rw-r--r-- root     system  /etc/vfs
-rw-r--r-- root     system  /etc/security/failedlogin
-rw-rw---- root     audit   /etc/security/audit/hosts
```

- Tiltsa le a root felhasználó távoli bejelentkezését. A root felhasználó csak a rendszerkonzolról jelentkezhessen be.
- Kapcsolja be a rendszerfigyelést. További tájékoztatást a következő részben talál: “Ellenőrzés áttekintése” oldalszám: 131.
- Engedélyezze a bejelentkezés-vezérlési irányelvet. További tájékoztatást a következő részben talál: “Bejelentkezés felügyelete” oldalszám: 33.
- Tiltsa meg a felhasználóknak az xhost parancs futtatását. További tájékoztatást a következő részben talál: “X11 és CDE problémák kezelése” oldalszám: 39.
- Akadályozza meg a **PATH** környezeti változó jogosulatlan módosítását. További tájékoztatást a következő részben talál: “PATH környezeti változó” oldalszám: 54.
- Tiltsa le a telnet, rlogin és rsh szolgáltatásokat. További tájékoztatást a következő részben talál: “TCP/IP biztonság” oldalszám: 201.
- Alakítson ki felhasználói fiókvezérlést. További tájékoztatást a következő részben talál: “Felhasználói fiók felügyelet” oldalszám: 51.
- Vezessen be szigorú jelszókezelési irányelveket. További tájékoztatást a következő részben talál: “Jelszavak” oldalszám: 62.
- Alakítson ki lemezterület-korlátozásokat a felhasználói fiókokhoz. További tájékoztatást a következő részben talál: “Kvótatúllépési helyzetek helyreállítása” oldalszám: 74.
- Csak a felügyeleti fiókok számára engedélyezze az **su** parancs használatát. Figyelje meg a **su** parancs naplóit a /var/adm/sulog fájlban.
- Engedélyezze a képernyő lezárását X Window használata esetén.
- Korlátozza a **cron** és **at** parancsok használatát azokra a fiókokra, amelyeknek valóban használniuk kell.
- Készítsen egy álnevet az **ls** parancsra, hogy megjelenítse a rejtett fájlneveket és a fájlnevek rejtett karaktereit.
- Készítsen egy álnevet az **rm** parancsra, hogy elkerülje a rendszer fájljainak véletlen törlését.
- Tiltsa le a szükségtelen hálózati szolgáltatásokat. További tájékoztatást a következő részben talál: “Hálózati szolgáltatások” oldalszám: 209.
- Mentse gyakran a rendszert és ellenőrizze a mentések integritását.
- Fizessen elő a biztonsággal kapcsolatos elektronikus levelezési listákra.

Az általános AIX rendszerszolgáltatások összefoglalása

Az alábbi táblázat összefoglalja az AIX legáltalánosabb rendszerszolgáltatásait. A táblázat segítségével azonosíthatja a rendszer biztonságossá tételéhez szükséges kiindulópontokat.

A rendszer biztonságossá tétele előtt mentse el az összes eredeti konfigurációs fájlt, különösen az alábbiakat:

- /etc/inetd.conf
- /etc/inittab
- /etc/rc.nfs
- /etc/rc.tcpip

Szolgáltatás	Démon	Indítás ideje	Funkció	Megjegyzések
inetd/bootps	inetd	/etc/inetd.conf	a lemez nélküli kliensek bootp szolgáltatásai	• A Hálózati telepítéskezeléshez (NIM) és a távoli rendszerbetöltéshez szükséges • A tftp-vel együtt működik • Tiltsa le a legtöbb esetben
inetd/chargen	inetd	/etc/inetd.conf	karaktergenerátor (csak teszteléshez)	• TCP és UDP szolgáltatásként is elérhető • Lehetőséget ad DoS típusú támadásokra • Tiltsa le, hacsak nem a hálózatot teszteli
inetd/cmsd	inetd	/etc/inetd.conf	naptárszolgáltatás (amit a CDE használ)	• Rootként fut, tehát biztonsági szempontból problémát jelenthet • Tiltsa le, hacsak nem szükséges a szolgáltatás a CDE miatt • Tiltsa le a háttér adatbázisszervereken
inetd/comsat	inetd	/etc/inetd.conf	Értesítés a beérkező elektronikus levelekről	• Rootként fut, tehát biztonsági szempontból problémát jelenthet • Ritkán van rá szükség • Tiltsa le
inetd/daytime	inetd	/etc/inetd.conf	elavult időszolgáltatás (csak teszteléshez)	• Rootként fut • TCP és UDP szolgáltatásként is elérhető • Lehetőséget ad DoS PING típusú támadásra • A szolgáltatás elavult, csak tesztelésre használatos • Tiltsa le
inetd/discard	inetd	/etc/inetd.conf	/dev/null szolgáltatás (csak teszteléshez)	• TCP és UDP szolgáltatásként is elérhető • DoS típusú támadásokban használatos • A szolgáltatás elavult, csak tesztelésre használatos • Tiltsa le
inetd/dtspc	inetd	/etc/inetd.conf	CDE alfolyamat-vezérlés	• Ezt a szolgáltatást az **inetd** démon indítja automatikusan, válaszul egy CDE kliens azon kérésére, hogy indítson el a rendszer egy folyamatot a démon hosztján. Éppen ezért támadás érheti ezt a szolgáltatást • Tiltsa le a háttérszervereken, amelyeken nem fut CDE • A CDE e funkció nélkül is működhet • Tiltsa le, kivéve, ha mindenképpen szükséges
inetd/echo	inetd	etc/inetd.conf	echo szolgáltatás (csak teszteléshez)	• TCP és UDP szolgáltatásként is elérhető • DoS és Smurf típusú támadásokban használható • A más helyen indított echóval a támadó a tűzfalon keresztüljuthat vagy adatvihart kavarhat • Tiltsa le
inetd/exec	inetd	/etc/inetd.conf	távoli végrehajtási szolgáltatás	• Root felhasználóként fut • Bekéri a felhasználói azonosítót és jelszót, amelyek titkosítás nélkül kerülnek továbbításra • Ez a szolgáltatás érzékeny a lehallgatásra • Tiltsa le

Szolgáltatás	Démon	Indítás ideje	Funkció	Megjegyzések
inetd/finger	inetd	/etc/inetd.conf	finger - felhasználók lekérdezése	• Root felhasználóként fut • Információt ad ki a rendszerről és a felhasználókról • Tiltsa le
inetd/ftp	inetd	/etc/inetd.conf	fájlátviteli protokoll	• Root felhasználóként fut • A felhasználói azonosító és a jelszó titkosítás nélkül kerül átvitelre, ezért lehallgatható • Tiltsa le a szolgáltatást és használjon helyette nyilvános, ingyenes biztonságos héj (SSH) programot
inetd/imap2	inetd	/etc/inetd.conf	Internetes levélhozzáférési protokoll	• Ellenőrizze, hogy a program legfrissebb verzióját használja a szerveren • Csak akkor van rá szükség, ha levélkezelő szervert üzemeltet. Ellenkező esetben tiltsa le • A felhasználói azonosító és a jelszó titkosítás nélkül kerül továbbításra
inetd/klogin	inetd	/etc/inetd.conf	Kerberos bejelentkezés	• Akkor van bekapcsolva, ha a rendszerben Kerberos hitelesítést használnak
inetd/kshell	inetd	/etc/inetd.conf	Kerberos héj	• Akkor van bekapcsolva, ha a rendszerben Kerberos hitelesítést használnak
inetd/login	inetd	/etc/inetd.conf	rlogin szolgáltatás	• Érzékeny az IP cím hamisítás és DNS cím hamisítás típusú támadásokra • Az adatok, így a felhasználói azonosító és a jelszó is, titkosítás nélkül kerül továbbításra • Root felhasználóként fut • Használjon SSH-t (biztonságos héjprogramot) e szolgáltatás helyett
inetd/netstat	inetd	/etc/inetd.conf	aktuális hálózati állapot jelentése	• A rendszeren lefuttatva esetleg információkat adhat ki a hálózatról a crackereknek • Tiltsa le
inetd/ntalk	inetd	/etc/inetd.conf	Lehetővé teszi a felhasználóknak, hogy csevegjenek egymással	• Root felhasználóként fut • Éles vagy háttérszervereken nincs szükség rá • Tiltsa le, kivéve, ha mindenképpen szükséges
inetd/pcnfsd	inetd	/etc/inetd.conf	PC NFS fájlszolgáltatások	• Tiltsa le a szolgáltatást, ha nem használja • Ha hasonló szolgáltatást keres, érdemes megfontolni a Samba használatát; a pcnfsd démon ugyanis még a Microsoft-féle SMB-specifikációk előttről származik

Szolgáltatás	Démon	Indítás ideje	Funkció	Megjegyzések
inetd/pop3	inetd	/etc/linetd.conf	Postahivatal protokoll	• A felhasználói azonosítók és jelszavak titkosítás nélkül kerülnek továbbításra • Csak akkor van rá szükség, ha a rendszer levélkezelő szerver és vannak olyan kliensek, amelyeken csak POP3-at kezelő alkalmazások futnak • Ha a kliensek képesek IMAP kezelésére, akkor inkább azt használja; illetve használja helyette a POP3s szolgáltatást. Ez utóbbi szolgáltatás Védett socket réteg (SSL) alagutat használ. • Tiltsa le, ha nem működtet levélkezelő szervert, illetve ha nincsenek POP szolgáltatást igénylő kliensek
inetd/rexd	inetd	/etc/inetd.conf	távoli végrehajtás	• Root felhasználóként fut • Az **on** parancs párja • Tiltsa le a szolgáltatást • Használja helyette az **rsh** és **rshd** szolgáltatásokat
inetd/quotad	inetd	/etc/inetd.conf	NFS kliensek fájlkvótáinak jelentése	• Csak akkor van rá szükség, ha NFS fájlszolgáltatásokat használ • Tiltsa le a szolgáltatást, hacsak nem kell választ adni a **quota** parancs kéréseire • Ha használnia kell ezt a szolgáltatást, akkor ügyeljen arra, hogy a szolgáltatás minden frissítését és javítását telepítse
inetd/rstatd	inetd	/etc/inetd.conf	Kernel statisztika szerver	• Ha figyelnie kell rendszereket, használjon inkább SNMP-t és tiltsa le ezt a szolgáltatást • Az **rup** parancs használatához szükséges
inetd/rusersd	inetd	/etc/inetd.conf	információk a bejelentkezett felhasználóról	• Ez nem lényeges szolgáltatás. Tiltsa le • Root felhasználóként fut • A rendszeren és a többi gépen kiadja az aktuális felhasználók listáját, és az rusers paranccsal egyenrangú
inetd/rwalld	inetd	/etc/inetd.conf	üzenet minden felhasználónak	• Root felhasználóként fut • Ha a rendszerben vannak interaktív felhasználók, akkor lehet, hogy meg kell tartani ezt a szolgáltatást • Éles vagy adatbázisszervereken semmi szükség rá • Tiltsa le
inetd/shell	inetd	/etc/inetd.conf	rsh szolgáltatás	• Hacsak lehetséges, tiltsa le ezt a szolgáltatást. Használjon inkább SSH-t. • Ha muszáj használni ezt a szolgáltatást, akkor a TCP Wrapper segítségével akadályozza meg a címhamisítást és korlátozza a veszélyeztetettséget • Az **Xhier** szoftverterjesztési program igényli
inetd/sprayd	inetd	/etc/inetd.conf	RPC spray tesztek	• Root felhasználóként fut • Szükség lehet rá az NFS hálózati problémáinak azonosításához • Tiltsa le, ha nem üzemeltet NFS-t

Szolgáltatás	Démon	Indítás ideje	Funkció	Megjegyzések
inetd/systat	inetd	/etc/inted.conf	"ps -ef" állapotjelentés	• Használatával távoli helyről is megtekinthető a rendszeren futó folyamatok állapota • A szolgáltatás alapértelmezettként le van tiltva. Időről időre ellenőrizze, hogy nem kapcsolta-e be valaki a szolgáltatást.
inetd/talk	inetd	/etc/inetd.conf	osztott képernyő a hálózat 2 felhasználója között	• Nem kötelező szolgáltatás • A **talk** parancs használja • UDP szolgáltatás az 517-es porton • Tiltsa le, hacsak nincs szükség több interaktív csevegési kapcsolatra a UNIX felhasználók számára
inetd/ntalk	inetd	/etc/inetd.conf	"új talk" - osztott képernyő a hálózat 2 felhasználója között	• Nem kötelező szolgáltatás • A **talk** parancs használja • UDP szolgáltatás az 517-es porton • Tiltsa le, hacsak nincs szükség több interaktív csevegési kapcsolatra a UNIX felhasználók számára
inetd/telnet	inetd	/etc/inetd.conf	telnet szolgáltatás	• Támogatja a távoli bejelentkezést, de a jelszó és az azonosító titkosítás nélkül kerül továbbításra • Hacsak lehetséges, tiltsa le ezt a szolgáltatást és használjon helyette SSH-t
inetd/tftp	inetd	/etc/inetd.conf	triviális fájlátvitel	• UDP szolgáltatás a 69-es porton • Rootként fut, tehát sérülékeny • A NIM használja • Tiltsa le, kivéve ha NIM-et használ vagy lemez nélküli munkaállomásokat kell távolról elindítania
inetd/time	inetd	/etc/inetd.conf	elavult időszolgáltatás	• Az **inetd** belső funkciója, amelyet az **rdate** parancs használ. • TCP és UDP szolgáltatásként is elérhető • Néha használják a rendszerórák szinkronizálására rendszerbetöltéskor • A szolgáltatás elavult. Használja helyette az ntpdate parancsot • Csak azután tiltsa le végleg, hogy kipróbálta a rendszert (indítás/újraindítás) a szolgáltatást letiltva és nem észlelt problémát
inetd/ttdbserver	inetd	/etc/inetd.conf	tool-talk adatbázisszerver (CDE-hez)	• Az **rpc.ttdbserverd** root felhasználóként fut, tehát sérülékeny • Kötelező szolgáltatásként van megjelölve CDE-hez, de valójában a CDE működik nélküle is • Háttérszervereken ne futtassa, se olyan szervereken, ahol a biztonság fontos szempont
inetd/uucp	inetd	/etc/inetd.conf	UUCP hálózat	• Tiltsa le, kivéve ha van olyan alkalmazás, amelyik használja az UUCP-t

Szolgáltatás	Démon	Indítás ideje	Funkció	Megjegyzések
inittab/dt	init	/etc/rc.dt parancsfájl az /etc/inittab könyvtárban	asztali bejelentkezés a CDE környezetbe	• Elindítja az X11 szervert a konzolon • Támogatja az X11 képernyőkezelő vezérlőprotokollt (xdcmp-t), hogy más X11 állomások is be tudjanak jelentkezni ugyanarra a gépre • A szolgáltatást csak személyes munkaállomásokon célszerű használni. Kerülje a háttérrendszereken használatot
inittab/dt_nogb	init	/etc/inittab	asztali bejelentkezés a CDE környezetbe (NEM grafikus indulás)	• Nem jelenik meg grafikus képernyő egészen addig, amíg a rendszer teljesen el nem indult • Ugyanazok, mint inittab/dt esetén
inittab/httpdlite	init	/etc/inittab	webszerver a **docsearch** parancshoz	• Alapértelmezett webszerver a docsearch alrendszerhez • Tiltsa le, kivéve, ha a gép dokumentációszerver
inittab/i4ls	init	/etc/inittab	licenckezelő szerverek	• Fejlesztési gépeken engedélyezze • Éles üzemi gépeken tiltsa le • Engedélyezze azokon a háttér adatbázisszervereken, amelyek licenckövetelményeket támasztanak • Támogatja a fordítókat, adatbázisszoftvert, illetve bármilyen egyéb licencelt termékeket
inittab/imqss	init	/etc/inittab	keresőmotor a "docsearch"-höz	• A docsearch alrendszer alapértelmezett webszerverének része • Tiltsa le, kivéve, ha a gép dokumentációszerver
inittab/lpd	init	/etc/inittab	BSD sornyomtató illesztő	• Nyomtatási feladatokat fogad más rendszerekből • Ha letiltja ezt a szolgáltatást, akkor is tud feladatokat küldeni a nyomtatószervernek • Tiltsa le, miután meggyőződött arról, hogy ez nincs kihatással a nyomtatásra
inittab/nfs	init	/etc/inittab	Hálózati fájlrendszer/ Hálózati információs szolgáltatások	• Az UDP/RPC-re épülő NFS és NIS szolgáltatások • Minimális mennyiségű hitelesítés • Tiltsa le a háttérrendszereken
inittab/piobe	init	/etc/inittab	nyomtató IO háttér (nyomtatáshoz)	• A **qdaemon** parancs által elküldött feladatok ütemezését, sorbaállítását és nyomtatását kezeli • Tiltsa le, ha nem nyomtat a rendszerről, mert a nyomtatási feladatokat egy szerverre küldi
inittab/qdaemon	init	/etc/inittab	várakozási sor démon (nyomtatáshoz)	• Nyomtatási feladatokat küld a **piobe** démonnak • Tiltsa le, ha nem nyomtat a rendszerről
inittab/uprintfd	init	/etc/inittab	kernelüzenetek	• Általában nincs rá szükség • Tiltsa le

Szolgáltatás	Démon	Indítás ideje	Funkció	Megjegyzések
inittab/writesrv	init	/etc/inittab	megjegyzések írása a tty-kre	• Csak interaktív UNIX munkaállomás felhasználók használják • Tiltsa le a szolgáltatást a szervereken, a háttér adatbázisokon, valamint a fejlesztési gépeken • Engedélyezze a szolgáltatást a munkaállomásokon
inittab/xdm	init	/etc/inittab	hagyományos X11 képernyőkezelés	• Ne futtassa éles háttér- vagy adatbázisszervereken • Ne futtassa fejlesztési rendszereken, kivéve, ha szükség van X11 képernyőkezelésre • Munkaállomásokon futhat, ha grafikára van szükség
rc.nfs/automountd		/etc/rc.nfs	gyorsítótáras fájlrendszerek	• NFS használata esetén engedélyezze a munkaállomásokon • Ne használja az automatikus beillesztőt fejlesztési vagy háttérszervereken
rc.nfs/biod		/etc/rc.nfs	Blokk IO démon (az NFS szerverhez szükséges)	• Csak az NFS szerveren engedélyezze • Ha nem NFS szerver, akkor tiltsa le az nfsd-vel és az rpc.mountd-vel együtt
rc.nfs/keyserv		/etc/rc.nfs	Biztonságos RPC kulcsszerver	• A biztonságos RPC-hez szükséges kulcsokat kezeli • Tiltsa le, ha *nem* használ NFS-t és NIS-t.
rc.nfs/nfsd		/etc/rc.nfs	NFS szolgáltatások (az NFS szerverhez szükséges)	• Gyenge hitelesítés • Származhat belőle verem keret összeomlás • Engedélyezze az NFS fájlszervereken • Ha letiltja, akkor tiltsa le a **biod**, **nfsd** és **rpc.mountd** szolgáltatásokat is
rc.nfs/rpc.lockd		/etc/rc.nfs	NFS fájlzárolások	• Tiltsa le, ha nem üzemeltet NFS-t • Tiltsa le, ha nem használ fájlzárolásokat a hálózaton • A **lockd** démon a SANS Tíz legnagyobb biztonsági veszély listájának egyike
rc.nfs/rpc.mountd		/etc/rc.nfs	NFS fájlbeillesztések (az NFS szerverhez szükséges)	• Gyenge hitelesítés • Származhat belőle verem keret összeomlás • Csak NFS fájlszervereken engedélyezze • Ha letiltja, akkor tiltsa le a **biod** és **nfsd** szolgáltatásokat is
rc.nfs/rpc.statd		/etc/rc.nfs	NFS fájlzárolások (a helyreállításukhoz)	• Fájlzárolások megvalósítása az NFS-ben • Tiltsa le, ha nem üzemeltet NFS-t
rc.nfs/rpc.yppasswdd		/etc/rc.nfs	NIS jelszódémon (az elsődleges NIS-hez)	• Használható a helyi jelszófájl manipulálására • Csak akkor van szükség rá, ha az adott gép az elsődleges NIS; minden egyéb esetben tiltsa le

Szolgáltatás	Démon	Indítás ideje	Funkció	Megjegyzések
rc.nfs/ypupdated		/etc/rc.nfs	NIS frissítési démon (alárendelt NIS-hez)	• Fogadja az elsődleges NIS-től érkező adatbázistérképeket • Csak akkor van szükség rá, ha az adott gép egy elsődleges NIS alárendeltje
rc.tcpip/autoconf6		/etc/rc.tcpip	IPv6 csatolók	• Tiltsa le, kivéve, ha IPv6-ot használ
rc.tcpip/dhcpcd		/etc/rc.tcpip	Dinamikus hoszt konfigurációs protokoll (kliens)	• Háttérszerverek nem szabad, hogy DHCP-t használjanak. Tiltsa le a szolgáltatást • Ha a hoszt nem használ DHCP-t, tiltsa le
rc.tcpip/dhcprd		/etc/rc.tcpip	Dinamikus hoszt konfigurációs protokoll (továbbító)	• Fogadja a szórt DHCP üzeneteket és továbbküldi egy másik hálózat szerverére • A szolgáltatás másodpéldánya megtalálható az útválasztókon • Tiltsa le, ha nem használ DHCP-t, vagy nem adja át a DHCP-adatokat más hálózatoknak
rc.tcpip/dhcpsd		/etc/rc.tcpip	Dinamikus hoszt konfigurációs protokoll (szerver)	• Megválaszolja a kliensek DHCP kéréseit azok indulásakor; információkat küld a kliensnek, például DNS-nevet, IP-címet, hálózati maszkot, útválasztó és üzenetszórási címeket • Tiltsa le, ha nem használ DHCP-t • Tiltsa le az éles üzemi és háttérszervereken, valamint az összes olyan gépen, amelyik nem használ DHCP-t
rc.tcpip/dpid2		/etc/rc.tcpip	elavult SNMP szolgáltatás	• Tiltsa le, kivéve, ha szüksége van az SNMP-re
rc.tcpip/gated		/etc.rc.tcpip	kapuzott útválasztás csatolók között	• Útválasztó-funkciót emulál • Tiltsa le a szolgáltatást és használjon helyette RIP-et vagy egy valódi útválasztót
rc.tcpip/inetd		/etc/rc.tcpip	inetd szolgáltatások	• Egy tökéletesen védett rendszeren le kéne tiltani, de ez általában gyakorlati okokból nem célszerű • A letiltás eredményeképpen leállnak a távoli héjszolgáltatások, amelyeket egyes levelező és webszerverek igényelnek
rc.tcpip/mrouted		/etc/rc.tcpip	multi-cast útvonalkezelés	• Útválasztó-funkciót emulál: multicast csomagokat küld az egyes hálózati szegmensek között • Tiltsa le a szolgáltatást. Használjon helyette valódi útválasztót
rc.tcpip/names		/etc/rc.tcpip	DNS névszerver	• Csak akkor használja, ha a gép egy DNS névszerver • Munkaállomásokon, fejlesztési és éles üzemi gépeken tiltsa le
rc.tcpip/ndp-host		/etc/rc.tcpip	IPv6 hoszt	• Tiltsa le, kivéve, ha IPv6-ot használ
rc.tcpip/ndp-router		/etc/rc.tcpip	IPv6 útválasztás	• Tiltsa le, kivéve, ha IPv6-ot használ. IPv6 helyett érdemes lehet útválasztót használni

Szolgáltatás	Démon	Indítás ideje	Funkció	Megjegyzések
rc.tcpip/portmap		/etc/rc.tcpip	RPC szolgáltatások	• Kötelező szolgáltatás • Az RPC szerverek a **portmap** démonnál jegyzik be magukat. Az RPC szolgáltatásokat kereső kliensek a **portmap** démontól kérdezik le, hol található egy adott szolgáltatás • Csak akkor tiltsa le, ha már annyira lecsökkentette az RPC szolgáltatásokat, hogy egyedül a **portmap** maradt.
rc.tcpip/routed		/etc/rc.tcpip	RIP útválasztás csatolók között	• Útválasztó-funkciót emulál • Tiltsa le, ha van útválasztó a csomagok hálózatok közötti forgalomirányítására
rc.tcpip/rwhod		/etc/rc.tcpip	Távoli "who" démon	• Összegyűjt és szétszór adatokat ugyanazon hálózat egyenrangú szervereinek • Tiltsa le a szolgáltatást
rc.tcpip/sendmail		/etc/rc.tcpip	levelezési szolgáltatások	• Root felhasználóként fut • Tiltsa le a szolgáltatást, kivéve, ha a gép levélkezelő szerverként működik • Ha le van tiltva, tegye a következőket: – Helyezzen el egy bejegyzést a crontab fájlban a sor kitörléséhez. Használja a **/usr/lib/sendmail -q** parancsot – Állítsa be a DNS szolgáltatásokat úgy, hogy a levelek másik rendszerhez kerüljenek továbbításra
rc.tcpip/snmpd		/etc/rc.tcpip	Egyszerű hálózatkezelési protokoll	• Tiltsa le, ha a rendszert nem figyeli SNMP eszközökkel • Az SNMP szükséges lehet kritikus fontosságú szervereken
rc.tcpip/syslogd		/etc/rc.tcpip	események rendszernaplója	• A szolgáltatás letiltása *nem* ajánlott • DoS típusú támadások érhetik • Minden rendszerben szükség van rá
rc.tcpip/timed		/etc/rc.tcpip	Régi idő démon	• Tiltsa le a szolgáltatást és használja helyette az xntp-t
rc.tcpip/xntpd		/etc/rc.tcpip	Új idő démon	• Szinkronizálja a rendszerek óráit • Tiltsa le a szolgáltatást. • Állítson be más rendszereket időszerverként és hagyja, hogy más rendszerek az ntpdate-et meghívó cron feladat segítségével szinkronizáljanak
dt login		/usr/dt/config/Xaccess	korlátozások nélküli CDE	• Ha nem biztosít CDE bejelentkezést X11 állomások csoportjainak, akkor korlátozhatja a dtlogin-t a konzolra.

Szolgáltatás	Démon	Indítás ideje	Funkció	Megjegyzések
anonim FTP szolgáltatás		user rmuser -p <felhasználónév>	anonim ftp	• Az anonim FTP nem teszi lehetővé annak megállapítását, melyik felhasználó használta az FTP-t • Törölje az ftp nevű felhasználót, ha van ilyen: **rmuser -p ftp** • Tovább növelhető a biztonság, ha az /etc/ftpusers fájlba beleírja azon felhasználók listáját, akik nem használhatják az ftp-t a rendszer elérésére
anonim FTP írások			anonim ftp feltöltések	• Semmilyen fájl ne tartozzon az ftp-hez. • Az anonim FTP feltöltések lehetővé teszik rosszindulatú kód elhelyezését a rendszeren. • Írja be a letiltani kívánt felhasználók nevét az /etc/ftpusers fájlba • Néhány példa, hogy mely, a rendszer által létrehozott felhasználóknak kívánja megtiltani az FTP-n keresztüli feltöltést: root, daemon, bin.sys, admin.uucp, guest, nobody, lpd, nuucp, ladp. • Módosítsa az ftpusers fájl tulajdonosi és csoportjogait: chown root:system /etc/ftpusers • Szigorítsa meg az ftpusers fájl engedélyét: chmod 644 /etc/ftpusers
ftp.restrict			ftp rendszerszámlákra	• Egyetlen külső felhasználó sem lehet képes rá, hogy az ftpusers fájl segítségével kicserélje a root fájlokat
root.access		/etc/security/user	rlogin/telnet a root azonosítóval	• Állítsa az etc/security/user fájlban az rlogin beállítást hamis értékre • Aki rootként akar bejelentkezni, tegye ezt először saját nevén, majd az **su** paranccsal váltson át rootra; ennek nyoma marad a megfigyelési naplóban
snmpd.readWrite		/etc/snmpd.conf	SNMP readWrite közösségek	• Ha *nem* használ SNMP-t, tiltsa le az SNMP démont. • Tiltsa le az /etc/snmpd.conf fájlban a private és system közösségeket • Korlátozza a "public" közösséget kizárólag azon IP címekre, amelyek figyelik a rendszert
syslog.conf			syslogd konfigurálása	• Ha nem állította be az /etc/syslog.conf fájlt, akkor tiltsa le ezt a démont • Ha használja a syslog.conf fájlt a rendszerüzenetek naplózására, akkor hagyja engedélyezve

Hálózati szolgáltatásbeállítások összefoglalása

Magasabb szintű rendszerbiztonság elérése érdekében több hálózati beállítás is van, amelyet a 0 beállítással letilthat, illetve az 1 megadásával engedélyezhet. A **no** paranccsal használható paramétereket a következő lista adja meg.

Paraméter	Parancs	Rendeltetés
bcastping	/usr/sbin/no -o bcastping=0	Engedélyezi az ICMP visszhang csomagokra adott választ az üzenetszórási címre. Letiltása megakadályozza a Smurf támadásokat.
clean_partial_conns	/usr/sbin/no -o clean_partial_conns=1	Megadásával elkerülhetők a SYN (sorozatszám szinkronizálási) támadások.
directed_broadcast	/usr/sbin/no -o directed_broadcast=0	Megadja, hogy megengedett-e az átjárónak szóló irányított üzenetszórás. 0-ra állítása megakadályozza, hogy az irányított csomagok távoli hálózatokba jussanak el.
icmpaddressmask	/usr/sbin/no -o icmpaddressmask=0	Megadja, hogy a rendszer válaszol-e az ICMP címmaszk kérésekre. Letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.
ipforwarding	/usr/sbin/no -o ipforwarding=0	Megadja, hogy a kernel továbbítsa-e a csomagokat. Letiltása megakadályozza, hogy az átirányított csomagok távoli hálózatokba jussanak el.
ipignoreredirects	/usr/sbin/no -o ipignoreredirects=1	Megadja, hogy a rendszer feldolgozza-e a kapott átirányításokat.
ipsendredirects	/usr/sbin/no -o ipsendredirects=0	Megadja, hogy a kernelnek küldenie kell-e átirányítási jelzéseket. Letiltása megakadályozza, hogy az átirányított csomagok távoli hálózatokba jussanak el.
ip6srcrouteforward	/usr/sbin/no -o ip6srcrouteforward=0	Megadja, hogy a rendszer továbbítja-e a forrás útválasztású IPv6 csomagokat. Letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.
ipsrcrouteforward	/usr/sbin/no -o ipsrcrouteforward=0	Megadja, hogy a rendszer továbbítja-e a forrás útválasztású csomagokat. Letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.
ipsrcrouterecv	/usr/sbin/no -o ipsrcrouterecv=0	Megadja, hogy a rendszer elfogadja-e a forrás útválasztású csomagokat. Letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.
ipsrcroutesend	/usr/sbin/no -o ipsrcroutesend=0	Megadja, hogy az alkalmazások küldhetnek-e forrás útválasztású csomagokat. Letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.
nonlocsroute	/usr/sbin/no -o nonlocsrcroute=0	Megadja, hogy az Internet protokollnak (IP), hogy szigorúan forrás útválasztású csomagok címezhetők a helyi hálózaton kívüli hosztokra is. Letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.
tcp_icmpsecure	/usr/sbin/no -o tcp_icmpsecurer=1	Védi a TCP kapcsolatokat ICMP (Internet vezérlőüzenet protokoll) forráscsillapítás és PMTUD (elérési út MTU feltérképezése) támadások ellen. Ellenőrzi az ICMP üzenet hasznos tartalmát annak megállapítása érdekében, hogy a TCP fejléc sorozatszáma az elfogadható tartományba esik-e. Értéket: 0=ki (alapértelmezett); 1=be.
ip_nfrag	/usr/sbin/no -o ip_nfrag=200	Az IP újraösszeállítási sorban egyidejűleg tárolható IP csomagtöredékek maximális számát adja meg (a 200 alapértelmezett érték egy IP csomag maximum 200 töredékét tárolja az IP újraösszeállítási sorban).
tcp_pmtu_discover	/usr/sbin/no -o tcp_pmtu_discover=0	Letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.

Paraméter	Parancs	Rendeltetés
tcp_tcpsecure	/usr/sbin/no -o tcp_tcpsecure=7	Védi a TCP kapcsolatokat támadásokkal szemben. Értékek: 0=nincs védelem; 1=hamis SYN küldése egy felépített kapcsolatnak; 2=hamis RST küldése egy felépített kapcsolatnak; 3=adatok injektálása egy felépített TCP kapcsolatba; 5–7=a fenti támadások kombinációja.
udp_pmtu_discover	/usr/sbin/no -o udp_pmtu_discover=0	Engedélyezi vagy letiltja az útvonal MTU feltérképezést a TCP alkalmazásokban. Letiltása megakadályozza a forrás útválasztással kapcsolatos támadásokat.

A hálózat beállítható paramétereiről további információkat a *Performance management* című kiadványban talál.

Trusted AIX

A Trusted AIX engedélyezi a többszintű biztonsági (MLS) képességeket AIX rendszeren.

Megjegyzés: Az MLS-t címke alapú biztonságnak is hívják.

A normál AIX rendszerrel összehasonlítva a Trusted AIX címke alapú biztonsága címkéket valósít meg a rendszer összes alanya és objektuma számára.

Megjegyzés: A Trusted AIX telepítési lehetősége lehetővé teszi a címke alapú biztonsággal rendelkező AIX környezetet. A rendszer hozzáférés-felügyelete az MLS környezethez biztosított címkékre épül és a következő támogatást is biztosítja:

- Címkézett objektumok: fájlok, IPC objektumok és más címkézett objektumok
- Címkézett nyomtatók
- Megbízható hálózat: RIPSO és CIPSO támogatása IPv4 és IPv6 protokollban

Ne felejtse el, hogy ha kiválasztja ezt a telepítési módot, akkor nem tud visszatérni a normál AIX környezethez a normál AIX felülírásos telepítésének végrehajtása nélkül. Mielőtt kiválasztja ezt a telepítési módot, értékelje ki a Trusted AIX környezet iránti igényét. A Trusted AIX rendszerrel kapcsolatos további részleteket az AIX nyilvánosan elérhető dokumentációja tartalmaz.

A szabványos AIX biztonsági szolgáltatások halmazát biztosítja, amelynek segítségével az információkezelők és adminisztrátorok alapszintű rendszer- és hálózati biztonságot biztosíthatnak. Az elsődleges AIX biztonsági szolgáltatások a következők:

- bejelentkezéssel és jelszóval vezérelt rendszer- és hálózat-hozzáférés
- felhasználói, csoport és mindenki más fájlhozzáférési jogosultságok
- hozzáférés felügyeleti listák (ACL)
- Megfigyelési alrendszer
- Szerep alapú hozzáférés-felügyelet (RBAC)

A Trusted AIX ezekre az elsődleges AIX operációs rendszer biztonsági szolgáltatásokra épül az AIX biztonság fokozása és hálózatkezelési alrendszerekre való kiterjesztése érdekében.

A Trusted AIX kompatibilis az AIX alkalmazás programozási felülettel (API). AIX rendszeren futó alkalmazások Trusted AIX rendszeren is futnak. A további biztonsági korlátozások miatt az MLS-re nem felkészített alkalmazásoknak jogosultságra lehet szükségük a Trusted AIX környezetben való működéshez. A **tracepriv** parancs segítségével az alkalmazások profilozhatók ilyen példahelyzetekben.

A Trusted AIX kiterjeszti az AIX alkalmazás programozási felületet a további biztonsági funkció támogatása érdekében. Ez lehetővé teszi, hogy az ügyfelek saját biztonságos alkalmazásokat fejlesszenek az AIX API és új Trusted AIX kiterjesztések segítségével.

A Trusted AIX lehetővé teszi, hogy az AIX rendszerek az információkat több biztonsági szinten dolgozzák fel. Ezt úgy alakították ki, hogy megfeleljen a US Department of Defense (DoD) TCSEC és European ITSEC kiterjesztett B1 biztonsággal kapcsolatos feltételeinek.

A szabványos AIX biztonsággal kapcsolatos információkért tekintse meg az Alap operációs rendszer biztonságossá tétele és Hálózat biztonságossá tétele című részt.

Trusted AIX bevezetése

A Trusted AIX kiterjeszti a szabványos AIX operációs rendszer biztonságát címke-alapú-biztonsági képességek szolgáltatásával az operációs rendszeren.

A Trusted AIX címke-alapú környezet a telepítési idejű lehetőségek kiválasztásával telepíthető. Trusted AIX telepítése esetén nem fog tudni visszatérni normál AIX környezethez a normál AIX felülírásos telepítésének végrehajtása nélkül. Telepítés után a Trusted AIX környezet a teljes AIX rendszerre alkalmazásra kerül, az AIX környezetben létrehozott WPAR-eket is beleértve. A címke alapú biztonságot (többszintű biztonságnak is hívják (MLS)) gyakran használják a hadi és védelmi iparágban, de a kereskedelmi iparágakban is használhatók. Ez a Trusted AIX rendszeren elérhető címkék személyre szabásával érhető el. A Trusted AIX friss telepítése lehetővé teszi olyan címékék használatát, amelyek szabványos MLS megvalósításokat követnek.

A Trusted AIX környezet normál AIX rendszerből és néhány további csomagból, illetve fájlkészletből áll. Ezen felül a kernelkapcsolók kényszerítik a kernelt Trusted AIX módban való működésre. CD-n vagy DVD-n keresztüli rendszerbetöltés esetén a rendszer normál AIX környezetben kerül betöltésre. A telepítési menük megjelenítésekor a telepítő kiválaszthatja a Trusted AIX lehetőséget és elindíthatja az MLS-sel kapcsolatos fájlok telepítését. A telepítés befejezése után a telepítőnek kezdeményeznie kell az első rendszerbetöltési szekvenciát. Az első rendszerbetöltési szekvencia során a konfigurációsegéd menüket biztosít a különböző felhasználók számára, és az ISSO, SA illetve SO felhasználók beállításra kerülnek; majd a rendszer befejezi a rendszerbetöltési művelet és ezzel az MLS kialakításra került.

A Trusted AIX kiterjeszti a rendszer biztonságát az információs biztonság négy elsődleges elemén keresztül:

- Bizalmasság
- Integritás
- Rendelkezésre állás
- Felelősségre vonhatóság

Az AIX által biztosított biztonsági szolgáltatásokon felül a Trusted AIX a következő képességeket biztosítja:

Érzékenységi címkék (SL-ek)
: Minden folyamat és fájl a biztonsági szintjének megfelelő címkét kap. A folyamatok csak a folyamat biztonsági tartományában lévő objektumokhoz férhetnek hozzá.

Integritás címkék (TL)
: Minden folyamat és fájl az integritási szintjének megfelelő címkét kap. A fájlnál alacsonyabb integritási szint címkével rendelkező folyamatok nem írhatják a fájlt. A folyamatok a náluk alacsonyabb integritási szint címkével rendelkező fájlokból nem olvashatnak.

Fájlok biztonsági kapcsolói
: Az egyéni fájlok további jelzőkkel rendelkezhetnek a műveletekkel kapcsolatos biztonság vezérlése érdekében.

Kernel biztonsági kapcsolók
: A teljes rendszer különböző engedélyezett és tiltott biztonsági szolgáltatásokkal rendelkezhet.

Jogosultságok
Számos parancs és rendszerhívás csak adott jogosultságokkal rendelkező folyamat számára áll rendelkezésre.

Felhatalmazások
Minden felhasználó egyedi felhatalmazáshalmazt kaphat. Minden felhatalmazás lehetővé teszi, hogy a felhasználó adott biztonsággal kapcsolatos funkciókat hajtson végre. A felhatalmazások szerepeken keresztül kerülnek felhasználóhoz rendelésre.

Szerepek
A szerep alapú hozzáférés-felügyeleti funkció a Trusted AIX részeként lehetővé teszi az adminisztrátori kötelességek szelektív átadását a nem root felhasználók számára. Ez az átadás az érintett felhatalmazások szerepbe gyűjtésével, majd ennek a szerepnek nem root felhasználóhoz rendelésével érhető el.

Bizalmasság

Az információk jogosulatlan felek számára történő biztosításával kapcsolatos fenyegetések bizalmassági problémát jelentenek.

A Trusted AIX objektum-újrafelhasználási és hozzáférés-felügyeleti mechanizmusokat biztosít az adaterőforrások védelme érdekében. Az operációs rendszer biztosítja, hogy a védett adaterőforrásokat csak adott felhatalmazással rendelkező felhasználók érhessék el és hogy ezek a felhasználók a védett erőforrásokat szándékosan és véletlenül se tehessék elérhetővé jogosulatlan felhasználók számára.

Az adminisztrátorok megakadályozhatják az érzékeny adatok hajlékonylemezre vagy egyéb cserélhető adathordozóra írását, nem védett nyomtatókon való kinyomtatását és hálózaton keresztül jogosulatlan távoli rendszerekre való átvitelét. Ezt a védelmet az operációs rendszer tartatja be és a rosszindulatú felhasználók vagy hamis folyamatok nem léphetik át.

Integritás

A jogosulatlan felek általi információmódosítással kapcsolatos fenyegetések integritási problémák.

A Trusted AIX számos biztonsági mechanizmust nyújt, amelyek biztosítják a Megbízható számítástechnikai alapkörnyezet és a védett adatok integritását attól függetlenül, hogy az adatok a rendszeren kerültek előállításra vagy importálva lettek hálózati erőforrásokon keresztül. A különböző hozzáférés-felügyeletei biztonsági mechanizmusok biztosítják, hogy csak a jogosult egyének módosíthassák az információkat. Annak megakadályozásához, hogy rosszindulatú felhasználók vagy hamis folyamatok birtokba vegyék vagy letiltsák a rendszererőforrásokat, a Trusted AIX megszünteti a root jogosultságot. Felhasználói root jogosultságok biztosítása helyett speciális adminisztrátori felhatalmazások és szerepek használata lehetővé teszi az adminisztrátori kötelességek szétválasztását.

Rendelkezésre állás

A szolgáltatások hosztgépen való elérhetőségével kapcsolatos fenyegetések elérhetőségi problémák. Ha például egy rosszindulatú program kitölti a fájlterületet és ezért nem hozható létre új fájl, akkor a hozzáférés továbbra is biztosított, de nincs rendelkezésre állás.

A Trusted AIX megvédi a rendszert jogosulatlan felhasználók és folyamatok elleni támadásoktól, amelyek a szolgáltatás megbénítását eredményezhetik. A jogosulatlan folyamatok nem olvashatják és nem írhatják a védett fájlokat, illetve könyvtárakat.

Felelősségre vonhatóság

Azzal kapcsolatos fenyegetések, hogy nem ismeretes, mely folyamatokat mely tevékenységek hajtják végre a rendszeren, felelősségre vonhatósági problémák. Ha például a rendszerfájlt megváltoztató felhasználó vagy folyamat nem követhető nyomon, akkor a jövőben nem határozható meg az ilyen tevékenységek leállításának módja.

Ez a kiterjesztett biztonsági szolgáltatás biztosítja az összes felhasználó azonosítását és hitelesítését, mielőtt a felhasználó hozzáférhetne a rendszerhez. A megfigyelési szolgáltatások az adminisztrátor számára megfigyelhető események halmazát és az összes szolgáltatás rendszeresemény nyomkövetését biztosítja.

Trusted AIX tulajdonságai

- A Trusted AIX az AIX telepítési menükkel kerül telepítésre. A Trusted AIX telepítése során további lehetőségek választhatók ki.
- A Trusted AIX környezet nem állítható vissza normál AIX környezetre a normál AIX környezet felülírásos telepítése nélkül.
- A root nem jelentkezhet be Trusted AIX környezetben.
- Trusted AIX környezetben a létrehozott WPAR-ek szintén címkézett biztonsági környezetben működnek.
- A Trusted AIX az MAC-t (kötelező hozzáférés-felügyelet) és MIC-t (kötelező integritásvezérlés) is támogatja. A kliens az MAC-hez és MIC-hez külálló címkehalmazokat adhat meg.
- A címkekódolási fájl az /etc/security/enc könyvtárban található és címke binárisra fordításával kapcsolatos információkat ment le. Az alapértelmezett címkekódolási fájl a CMW címkével kapcsolatos elnevezési követelményeket követi.
- A NIM telepítés kliensről való kezdeményezése támogatott. A szerverről átadott NIM telepítés nem lehetséges, mivel a root nem jelentkezhet be MLS rendszereken.
- A JFS2 (J2) fájlrendszer (kiterjesztett attribútumok 2-es változatának használata) fel lett készítve a címkék AIX rendszeren való tárolására. Más fájlrendszerek (mint például a J1 vagy NFS) Trusted AIX környezetben csak egyszintű fájlrendszerként (felépítési ponthoz rendelt címke) építhetők fel.
- Az X környezet Trusted AIX esetén le van tiltva.
- A Trusted AIX támogatja a CIPSO és RIPSO protokollt hálózati címke alapú kommunikáció esetén. Ezeket a protokollok az IPv4 és IPv6 egyaránt támogatja.
- Néhány AIX biztonsági mechanizmus közös a normál AIX és Trusted AIX rendszeren. Ezen közös biztonsági mechanizmusok közül kettő a szerep alapú hozzáférés-felügyelet (RBAC) és a megbízható végrehajtás az integritásellenőrzéshez.
- Mivel a root telepített Trusted AIX esetén le van tiltva, a telepítőnek be kell állítania jelszót az ISSO, SA és SO felhasználó számára a telepítési utáni első rendszerbetöltés során. A rendszer használhatatlan marad mindaddig, amíg nem hozza létre ezeket a jelszavakat.
- Az AIX 6 biztonsági szolgáltatások Redbooks kiadványa tartalmaz eseteket és példákat Trusted AIX rendszerhez.

Többszintű biztonság

A biztonságos rendszerek fő célja, hogy a telephely biztonsági házirendjének foganatosításával felelősségre vonhatóságot és elérhetőséget biztosítson.

A Trusted AIX biztonsági házirend egy meghatározott szabályhalmazt biztosít, amely meghatározza a megengedett rendszerhozzáférés-típusokat. Ez magában foglalja, hogy a felhasználók tevékenységükért felelősségre vonhatók, illetve megakadályozza az operációs rendszer módosítását.

A Trusted AIX a hozzáférés-felügyelet és az adott korlátozott hozzáférési feltételek segítségével vezérli a fájlok, könyvtárak, folyamatok és eszközök elérését.

A Trusted AIX az összes biztonsággal kapcsolatos eseményről nyomkövetési naplót tart fenn. A nyomkövetési napló köszönhetően az egyének felelősségre vonhatók, még a hatályos és valós felhasználói azonosítót módosító (például **su** parancs) programok esetében is. A Trusted AIX ezen kívül az adminisztrációs funkciókat felhatalmazásokkal és a legkevesebb jogosultság használatával (a művelet végrehajtását a felhasználó számára lehetővé tevő legkevesebb felhatalmazás biztosításával) adott egyénekre korlátozza.

Azonosítás és hitelesítés

Az azonosítás és hitelesítés (I&A) biztonsági mechanizmusok felelősek annak biztosításáért, hogy a rendszerhez hozzáférést kérő minden egyén megfelelően azonosításra és hitelesítésre kerül. Az azonosítás felhasználónevet, a hitelesítés jelszót igényel.

Minden Trusted AIX fiók jelszóval védett. Az információrendszer adatvédelmi megbízott (ISSO) a rendszert beállíthatja úgy, hogy a felhasználók számára lehetővé tegye saját jelszavuk kiválasztását, a jelszóhossz és összetettségi megszorítások figyelembevételével. Az ISSO ezen kívül a felhasználók szintjén meghatározhatja a minimális és maximális jelszó-öregedési paramétereket is (elévülési időtartamot), beleértve a jelszó elévülése előtti figyelmeztetési időszakokat is.

Az azonosítási és hitelesítési biztonsági mechanizmusok megkövetelik, hogy minden felhasználónév és felhasználói azonosító egyedi legyen. Az érvényes jelszóval nem rendelkező fiókok bejelentkezéshez nem használhatók. Az új felhasználók számára az ISSO szereppel rendelkező felhasználónak kell a kezdeti jelszót felvenni. Minden felhasználóhoz hozzárendelésre kerül egy egyedi azonosító, amelyet a rendszer megfigyelési célokra használ.

A jelszó csak titkosított formátumban kerül tárolásra. A jelszavak a rendszeren sima szöveg formátumban nem kerülnek tárolásra. A titkosított jelszavakat az árnyék jelszófájl tartalmazza, amelyhez csak a privilegizált folyamatok férhetnek hozzá. További információkat a **passwd** parancs leírása tartalmaz.

A Trusted AIX rendszerek két típusú fiókot ismernek fel: a rendszer- és a felhasználói fiókokat. A rendszerfiókok a 128-as felhasználói azonosítónál kisebbel rendelkező fiókok. Bár a rendszerfiókokhoz tartozhat jelszó, ezek a fiókok a rendszerre való bejelentkezés során nem használhatók.

Tetszés szerinti hozzáférés-felügyelet

A tetszés szerinti hozzáférés-felügyelet (DAC) a biztonság azon tényezője, amely a fájl- vagy könyvtártulajdonos felügyelete alatt áll.

UNIX jogosultságok

Az erőforrásra tulajdonosi hozzáféréssel rendelkező felhasználók az alábbiakat végezhetik el:

- Egyéb felhasználóknak közvetlenül hozzáférést adhatnak
- Egyéb felhasználóknak egy másolathoz hozzáférést adhatnak
- Az eredeti erőforráshoz hozzáférést lehetővé tevő programot biztosíthatnak (például a SUID programok segítségével)

A hagyományos UNIX jogosultsági bit módszer (tulajdonos/csoport/egyéb, illetve olvasás/írás/végrehajtás) szintén erre a DAC funkcionalitásra ad példát.

A jogosultsági bitek segítségével a felhasználók a fájlban található adatokhoz való hozzáférést a felhasználók és csoportok számára biztosíthatják, illetve letilthatják (a korlátozott hozzáférés feltétel alapján). A hozzáféréstípus alapjául a felhasználói azonosító és azok a csoportok szolgálnak, amelynek a felhasználó tagja. Minden fájlrendszer-objektum társított jogosultságokkal rendelkezik, amelyek leírják a tulajdonos, a csoport, illetve mindenki más hozzáférését.

A fájl tulajdonosa ezen kívül egyéb felhasználók számára úgy is adhat hozzáférést, hogy a **chown** és **chgrp** parancsok segítségével a fájl tulajdonjogát vagy csoportját módosítja

umask

A fájl létrehozásakor kezdetben az összes jogosultsági bit beállított. A bejelentkezési folyamat során beállított umask folyamat ezt követően eltávolítja a fájl bizonyos jogosultsági bitjeit. Az alapértelmezett umask a felhasználó parancsértelmezője által létrehozott minden fájlra, illetve a felhasználó parancsértelmezőjéből futtatott minden parancsra vonatkozik.

Alapértelmezésben a kernelelemek umask beállítása 000 (amely az összes jogosultságot elérhetővé teszi a felhasználók számára). Az AIX az umask beállítást 022 értékre állítja be (amely a csoport és mindenki más írási jogosultsági bitjeit kikapcsolja). Szükséges esetén azonban a felhasználók ezt a beállítást felülbírálhatják.

Megjegyzés: Az umask beállítást a 022-nél engedékenyebb értékre nagyon óvatosan módosítsa. Ha a fájlokon és folyamatokon több jogosultság áll rendelkezésre, akkor maga az egész rendszer kevésbé biztonságossá válhat.
Az alapértelmezett umask beállítás felülbírálására két módszer létezik:

- Az umask értékek a felhasználói .profile, .login vagy .chsrc fájlban módosíthatók. A módosítások a bejelentkezési munkamenet során létrehozott minden fájlt érintik.
- Az egyéni folyamatok umask szintjét az **umask** parancs segítségével állíthatja be. Az **umask** parancs futtatása után az új umask érték érinti a létrehozott összes új fájlt, amíg a következő két esemény közül az egyik be nem következik:
  - Az **umask** parancsot ismét nem futtatja

    VAGY
  - Kilép a parancsértelmezőből, amelyben az **umask** parancs kiadásra került

Ha az **umask** parancsot argumentumok nélkül futtatja, akkor az **umask** parancs visszaadja a munkamenet aktuális umask értékét.

Ajánlott engedélyezni a bejelentkezési munkamenetnek a kernel 022 umask értékének öröklését azáltal, hogy a profilokban umask értéket nem ad meg. A 022-nél kevésbé szigorú umask értékeket csak nagy körültekintéssel szabad használni.

Ha bizonyos fájlokhoz további jogosultságok szükségesek, akkor ezeket a **chmod** parancs körültekintő használatával, a fájl létrehozását követően tanácsos beállítani.

Hozzáférés felügyeleti listák

A szabványos UNIX jogosultsági bitek és az umask értéken kívül az AIX támogatja a hozzáférés felügyeleti listák használatát is.

A UNIX jogosultsági bitek a hozzáférést csak a fájltulajdonos, egy csoport, illetve a rendszer összes felhasználója számára felügyelik. ACL használatával a fájltulajdonos további felhasználóknak és csoportoknak is meghatározhat hozzáférési jogokat. A jogosultsági bitekhez hasonlóan a hozzáférés felügyeleti listák egy-egy rendszerobjektumhoz tartoznak (például egy fájlhoz vagy könyvtárhoz).

setuid és setgid jogosultsági bitek

A setuid és setgid jogosultsági bitek (felhasználói azonosító beállítása, illetve csoportazonosító beállítása) segítségével a programfájlok a programot futtató személy felhasználói és csoportazonosítója helyett a tulajdonos felhasználói és csoportazonosítójával futtathatók. Ez a fájlhoz tartozó setuid és setgid bitek beállításával érhető el. Ezáltal lehetővé válik az olyan védett alrendszerek fejlesztése, amelyben a felhasználók bizonyos fájlokhoz úgy férhetnek hozzá, illetve bizonyos fájlokat úgy futtathatnak, hogy közben nem a fájl tulajdonosai.

Ha a setgid bit az objektum létrehozásakor a szülőkönyvtáron beállított, akkor az új objektum a szülőkönyvtárral - nem pedig az objektum létrehozójával - egyező csoporthoz fog tartozni. Azonban a beállított setuid bittel rendelkező könyvtárban létrehozott objektumok tulajdonosa az objektum létrehozója, nem pedig a könyvtár tulajdonosa. Alkönyvtár létrehozásakor a szülőkönyvtár setuid/setgid bitjeit az alkönyvtárak öröklik.

A setuid és setgid jogosultsági bitek potenciális biztonsági kockázati tényezők. Az olyan programok, amelyek tulajdonosa a futtatás során a root felhasználó, tulajdonképpen korlátlan hozzáféréssel rendelkezhetnek a rendszer felett. A Trusted AIX rendszereken azonban a jogosultságok és egyéb hozzáférés-felügyeletek használata jelentősen csökkentik ezt a biztonsági kockázati tényezőt.

Szerep alapú hozzáférés-felügyelet elemei

A Trusted AIX támogatja a szerep alapú hozzáférés-felügyeletet (RBAC). Az RBAC egy olyan operációs rendszer mechanizmus, amelynek köszönhetően a hozzájuk rendelt szerepek segítségével a rootra, illetve felettes felhasználóra jellemző funkciókat normál felhasználók is végrehajthatják.

Az AIX RBAC központi elemei:

Felhatalmazások
A karaktersorozatok jelölik az általuk képviselt és név szerint közvetlenül vezérelt jogosultsági műveletet. Például az aix.network.manage felhatalmazási karaktersorozat a hálózatfelügyeleti funkciót határozza meg az AIX operációs rendszerben.

Jogosultságok
A jogosultság egy folyamatattribútum, amely lehetővé teszi, hogy a folyamat átlépjen adott rendszermegszorításokat és -korlátozásokat. A folyamathoz jogosultságok kerülnek hozzárendelésre, amelyek jellemzően egy privilegizált parancs meghívásán keresztül kerülnek megszerzésre.

Szerepek
Az AIX RBAC szerepelemei lehetővé teszik a felhasználók számára, hogy a rendszer kezelési funkcióinak egy részét egyesítsék, majd a funkciókat - kezelés céljából - egy normális felhasználóhoz rendeljék. Az AIX szerepek a felhatalmazások (mind rendszer-, mind egyéni felhatalmazások) gyűjteményéből, illetve tetszőleges egyéb szerepekből (például alszerepekből) állnak.

A szerep alapú hozzáférés-felügyelettel kapcsolatosan további információkat az RBAC témakör tartalmaz.

Kötelező hozzáférés-felügyelet

A kötelező hozzáférés-felügyelet az objektum-hozzáférés korlátozásának rendszer által foganatosított módszere, az objektum érzékenysége és a felhasználó engedélye alapján. Ezzel szemben a tetszés szerinti hozzáférés-felügyeletet az egyedi fájlok tulajdonosai - és nem a rendszer - foganatosítják.

MAC címkék használata

A MAC foganatosításához a Trusted AIX címkerendszert használ. A Trusted AIX rendszeren minden nevesített objektum érzékenységi címkével (SL) rendelkezik, amely azonosítja az objektum érzékenységi szintjét. A folyamatok szintén rendelkeznek SL címkével. A folyamat SL címkék azt jelölik, hogy a folyamatok milyen szintű érzékenységgel rendelkező információkhoz férhetnek hozzá. Általánosságban a folyamatoknak egy objektum eléréséhez az objektuméval egyenlő vagy annál nagyobb érzékenységi szinttel kell rendelkezniük. Az SL címkék segítségével a fájlok a normális felhasználók számára csak olvasható módon elérhetővé, illetve teljesen elérhetetlenné tehetők.

Minden rendszerobjektum - fájlok, IPC objektumok, hálózati kapcsolatok, folyamatok - rendelkezik SL címkével. Az objektumok létrehozásukkor automatikusan SL címkét kapnak. Minden tárkiíratás objektumnak számít, tehát a rendszer ezeket is automatikusan címkével látja el.

A Trusted AIX telepítése előtt a rendszeren található objektumok az alapértelmezett SYSTEM_LOW SL (SLSL) címkét kapják meg akkor, amikor az objektumokhoz a Trusted AIX telepítése után hozzáférés történik. Ezeken az objektumokon az SL beállítása nem végleges. Az objektum SL címkéjének beállításához futtassa a **settxattr** parancsot. A Trusted AIX telepítését követően létrehozott objektumok esetében az objektum SL címkéjét a rendszer a létrehozó folyamat SL címkéjére állítja be.

Felhasználók és címkék

A rendszer minden felhasználói fiókhoz hozzárendel egy érvényes SL-tartományt - a rendszer alapértelmezéseken vagy a felhasználóra jellemző beállításokon keresztül -, és a felhasználó csak ezen a tartományon belül dolgozhat. A folyamatok és felhasználók csak aktuális érzékenységi címkéjükkel hozhatnak létre fájlokat és könyvtárakat, illetve fájlokat csak a rendszer által foganatosított MAC megszorításoknak megfelelően olvashatnak és írhatnak.

MAC foganatosítása

A kötelező hozzáférés-felügyelet (MAC) mindig foganatosításra kerül, amikor egy folyamat kísérletet tesz egy fájlrendszerobjektum megnyitására, egy fájlrendszerobjektum attribútumainak lekérésére, egy folyamat felé jelzés küldésére, adatok adatfolyamon keresztül történő átvitelére, illetve csomag küldésére vagy fogadására egy hálózati csatolón keresztül. A fájlrendszer objektumai csak akkor érhetők el, ha a MAC és DAC feltételek egyaránt teljesülnek. Amikor egy felhasználó megpróbál egy fájlhoz hozzáférni, akkor a MAC megszorítások a DAC megszorítások (például a jogosultsági bitek vagy ACL listák) ellenőrzése előtt kerülnek foganatosításra.

A fájlrendszerobjektumok elérését nem csak az objektum SL címkéje, de az objektumot tároló könyvtár SL címkéje is korlátozza. Ennek következtében a fájlrendszerobjektumok védhetők az objektum saját SL címkéjétől eltérő érzékenységi szinten is (a könyvtár SL címkéjén). A fájlrendszerobjektumok több könyvtárban elhelyezett névvel (hivatkozással) rendelkezhetnek. Ugyan az összes nevet (hivatkozást) a rendszer a hivatkozás céljaként megadott fájllal egyező SL címkével védi, a különböző hivatkozások tényleges védelme eltérő lehet, mivel a hivatkozások eltérő szintű védelemmel rendelkező könyvtárakban találhatók.

Az objektum neve az objektumot tartalmazó könyvtárban kerül tárolásra. Tehát az adott könyvtárhoz hozzáféréssel rendelkező összes folyamat képes a könyvtárban található összes objektum nevének megjelenítésére. Azonban az objektumokat csak a megfelelő hozzáféréssel rendelkező folyamatok írhatják, illetve olvashatják.

SL címkék listázása és módosítása

A rendszer objektumainak és folyamatainak SL címkéje a **lstxattr** parancs segítségével jeleníthető meg, illetve a **settxattr** parancs segítségével módosítható.

A fájlok és folyamatok SL címkéjét csak a megfelelő jogosultsággal rendelkező folyamatok és megfelelő felhatalmazással rendelkező felhasználók módosíthatják.

Ahhoz, hogy a felhasználók a **settxattr** paranccsal egy fájlrendszerobjektum SL címkéjét alacsonyabb szintű SL címkére módosíthassa, a felhasználónak aix.mls.label.sl.downgrade felhatalmazással kell rendelkeznie. A fájlrendszerobjektumok SL címkéjének kiemeléséhez a felhasználónak aix.mls.label.sl.upgrade felhatalmazással kell rendelkeznie. A folyamatok SL címkéjének módosítása esetében a kiemeléshez a felhasználónak aix.mls.proc.sl.upgrade felhatalmazással, a visszaléptetéshez aix.mls.proc.sl.downgrade felhatalmazással kell rendelkeznie.

MAC nyitott fájlleírókon

Az írási/olvasási és egyszerű fájlhozzáférések esetén a MAC ellenőrzések akkor kerülnek végrehajtásra, amikor a folyamat a fájlhoz hozzáfér. Miután a folyamat már rendelkezik a fájl leírójával, a fájlt olvashatja és írhatja, még akkor is, ha a folyamat SL címkéje a fájl SL címkéjénél alacsonyabb szintűre változik. Azonban bizonyos műveletek (például a fájl tulajdonosának, engedélyeinek, címkéinek és jogosultságainak beállítása) a hozzáférési ellenőrzéseket azt követően hajtják végre, hogy a folyamat a fájlleírót megszerezte.

Más szóval, a MAC ellenőrzések és a particionált könyvtárútvonal-feloldások nem akkor kerülnek végrehajtásra, amikor egy folyamat a fájlt a fájlleíró segítségével eléri. A fájl és/vagy folyamat SL címkéje változhat, de a hozzáférés továbbra is engedélyezett.

Kötelező integritásfelügyelet

A kötelező integritásfelügyelet az objektum-hozzáférés és -módosítás korlátozásának rendszer által foganatosított módszere, az objektum integritása és a felhasználó engedélye alapján. Míg a MAC az objektum érzékenységével kapcsolatos, addig a MIC az objektum megbízhatóságát érinti.

MIC címkék használata

A MIC foganatosításához a Trusted AIX címkerendszert használ. A Trusted AIX rendszeren minden nevesített objektum integritás címkével (TL) rendelkezik, amely azonosítja az objektum integritás szintjét. A folyamatok szintén

rendelkeznek TL címkével. A folyamat TL címkék jelölik, hogy a folyamat milyen szintű információs integritáshoz férhet hozzá. Minél magasabb a TL, az objektum vagy folyamat annál megbízhatóbb.

Az objektum módosításához a folyamatnak legalább ugyanolyan megbízhatónak kell lennie, mint magának az objektumnak. Ennek következtében a folyamatoknak az objektuméval egyenlő vagy annál nagyobb TL címkével kell rendelkezniük. Ennek következtében az integritás címkék segítségével a fájlok csak olvasható módon tehetők elérhetővé.

Ezen kívül a folyamatok nem használhatnak magánál a folyamatnál kevésbé megbízható objektumokból származó adatokat. Ennek következtében az objektumoknak a folyamatéval egyenlő vagy annál nagyobb TL címkével kell rendelkezniük.

Minden rendszerobjektum - például fájlok és folyamatok - rendelkezik TL címkével. Az objektumok létrehozásukkor automatikusan TL címkét kapnak. Minden tárkiíratás objektumnak számít, tehát a rendszer ezeket is automatikusan címkével látja el.

A Trusted AIX telepítése előtt a rendszeren meglévő objektumok az alapértelmezett SYSTEM_LOW TL (SLTL) címkét kapják meg akkor, amikor az objektumokhoz a Trusted AIX telepítése után hozzáférés történik. Ezeken az objektumokon a TL beállítása nem végleges. Az objektum TL címkéjének beállításához futtassa a **settxattr** parancsot. A Trusted AIX telepítését követően létrehozott objektumok esetében az objektum TL címkéjét a rendszer a létrehozó folyamat TL címkéjére állítja be.

Felhasználók és címkék

A rendszer minden felhasználói fiókhoz hozzárendel egy érvényes TL-tartományt - a rendszer alapértelmezéseken vagy a felhasználóra jellemző beállításokon keresztül -, és a felhasználó csak ezen a tartományon belül dolgozhat. A folyamatok és felhasználók csak aktuális TL címkéjükkel hozhatnak létre fájlokat és könyvtárakat, illetve fájlokat csak a rendszer által foganatosított MIC megszorításoknak megfelelően olvashatnak és írhatnak.

MIC foganatosítása

A kötelező integritásellenőrzés (MIC) a MAC foganatosításakor mindig foganatosításra kerül. Ezen kívül a MIC foganatosításra kerül fájlok és könyvtárak törlésekor vagy átnevezésekor is.

TL címkék módosítása

Az objektumok és folyamatok TL címkéje az **lstxattr** parancs segítségével jeleníthető meg, illetve a **settxattr** parancs segítségével módosítható.

A fájlok és folyamatok TL címkéjét csak a megfelelő jogosultsággal rendelkező folyamatok és megfelelő felhatalmazással rendelkező felhasználók módosíthatják. Ahhoz, hogy a felhasználó a **settxattr** paranccsal egy fájlrendszerobjektum TL címkéjét alacsonyabb szintű TL címkére módosíthassa, aix.mls.label.tl.downgrade felhatalmazással kell rendelkeznie. A fájlrendszerobjektumok TL címkéjének kiemeléséhez a felhasználónak aix.mls.label.tl.upgrade felhatalmazással kell rendelkeznie. A folyamatok TL címkéjének módosítása esetében a kiemeléshez a felhasználónak aix.mls.proc.tl.upgrade felhatalmazással, a visszaléptetéshez pedig aix.mls.proc.tl.downgrade felhatalmazással kell rendelkeznie.

NOTL

Létezik egy különleges TL is, az ún. NOTL, amely fájlrendszerekre, IPC objektumokra és folyamatokra alkalmazható. Ha egy objektum vagy folyamat NOTL TL címkével rendelkezik, akkor az adott objektumon vagy folyamaton a rendszer MIC ellenőrzést nem végez. A TL címkét NOTL értékre, illetve a jelenleg NOTL beállítású TL címkéket egyébre csak privilegizált felhasználók módosíthatják.

MIC nyitott fájlleírókon

Az írási/olvasási és egyszerű fájlhozzáférések esetén a MIC ellenőrzések akkor kerülnek végrehajtásra, amikor a folyamat a fájlhoz hozzáfér. Miután a folyamat már rendelkezik a fájl leírójával, a fájlt olvashatja és írhatja, még akkor is, ha a folyamat TL címkéje a fájl TL címkéjénél alacsonyabb szintűre változik. Azonban bizonyos műveletek (például a fájl tulajdonosának, engedélyeinek, címkéinek és jogosultságainak beállítása) a hozzáférési ellenőrzéseket azt követően hajtják végre, hogy a folyamat a fájlleírót megszerezte. Más szóval, a MIC ellenőrzések nem akkor kerülnek végrehajtásra, amikor egy folyamat a fájlt a fájlleíró segítségével eléri. A fájl és/vagy folyamat TL címkéje változhat, de a hozzáférés továbbra is engedélyezett.

Címkék

A Trusted AIX rendszereken az alanyok és objektumok biztonsági szintjét címkék jelölik. A rendszeren használatos címkéket, illetve a közöttük fennálló kapcsolatot az ISSO határozza meg.

Érzékenységi címkék (SL-ek):

Az egyes alanyokhoz és objektumokhoz tartozó SL-ek kötelező hozzáférés-felügyeleti irányelvet foganatosítanak a hozzáférés-felügyelet Bell-LaPadula modellje alapján.

Az SL két részből áll:

- Hierarchikus osztályozás
- Egy vagy több szakasz halmaza

Minden telepítési hely megadhatja a rendszer címkéinek nevét és kapcsolatát. A rendszeradminisztrátor be tudja állítani a neveket és a kapcsolatokat a címkekódolási fájlokban lévő helyirányelvek által előírt módon.

SL osztályozások:

Az osztályozások hierarchikusan rendezettek, és egy-egy érzékenységi szintet képviselnek.

Ha például a telephely érvényes osztályozásai Top Secret, Secret és Unclassified, akkor a Top Secret érzékenyebb a Secret osztályozásnál, illetve a Secret érzékenyebb az Unclassified osztályozásnál. A Trusted AIX legfeljebb 32 000 hierarchikus besorolás használatát támogatja.

SL szakszok:

A szakaszok témaköröket vagy munkacsoportokat képviselnek. Minden szakasz saját névvel rendelkezik (például: NATO vagy CRYPTO).

A szakaszok belső logikájuk alapján nem rendezettek, de az ISSO megszorításokat hozhat létre arra vonatkozóan, hogy mely szakaszok és besorolások egyesíthetők. A Trusted AIX legfeljebb 1 024 szakasz használatát támogatja.

SL összetevők:

Felhasználó által olvasható formátumban az SL-t elemek karaktersorozata ábrázolja. Az első elem az osztályozást ábrázolja, a többi pedig a szakaszokat. Az elemek szóközzel vannak elválasztva.

A brazil gazdasággal kapcsolatos szigorúan bizalmas információkat tartalmazó fájlnak például a hierarchikus osztályozása szigorúan bizalmas (TS), a szakasz pedig a brazil (B) és gazdaság (e) lehet. Az SL felhasználó által olvasható formátuma TS B e vagy Top Secret Brazil economy lenne.

SL kapcsolatok:

A rendszerfelhasználónak meg kell értenie a címkék és a címkék használata közötti kapcsolatot.

A MAC címkék között háromféle kapcsolat lehet:

- Dominancia
- Egyenlőség
- Nem összehasonlítható

Dominancia

Egy SL (L1) csak akkor domináns a másikkal szemben (L2), ha a következő feltételek teljesülnek:
- Az L1 osztályozása megegyezik az L2 osztályozásával, vagy meghaladja azt
- Az L1 szakaszainak halmaza tartalmazza az L2 teljes szakaszhalmazát

Feltételezzük például az L1 szigorúan bizalmas információkat tartalmazó SL-t A és B szakaszokkal (TS A B), és másik L2 titkos információkat tartalmazó SL-t A szakasszal, B nélkül (S A). A TS A B domináns az S A SL-lel szemben, mivel a TS osztályozás domináns az S osztályozással szemben, és az L1 szakaszainak halmaza teljes egészében tartalmazza a L2 szakaszainak halmazát. Ebben a példában az L2 nem domináns az L1 SL-lel szemben.

34. táblázat: SL dominancia

L1		L2		Dominancia
Címke	**Szakasz**	**Címke**	**Szakasz**	
TOP SECRET	A,B	SECRET	A	L1 > L2

Egyenlőség

Egy SL (L1) csak akkor egyenlő a másikkal (L2), ha a következő feltételek teljesülnek:
- Az L1 osztályozása megegyezik az L2 osztályozásával
- Az L1 szakaszainak halmaza megegyezik az L2 szakaszhalmazával

Ha két címke azonos, akkor minden címke domináns a másikkal szemben. Feltételezzük például a szigorúan bizalmas információkkal rendelkező fájlhoz tartozó az SL-t A (TS A) szakasszal, és egy másik szigorúan bizalmas információkat tartalmazó fájlt A szakasszal (szintén TS A). Az SL-ek megegyeznek és dominánsak egymással szemben.

35. táblázat: SL egyenlőség

L1		L2		Dominancia
Címke	**Szakasz**	**Címke**	**Szakasz**	
TOP SECRET	A	TOP SECRET	A	L1 = L2

Nem összehasonlítható

Két SL nem összehasonlítható, ha (L1 nem egyenlő L2-vel, és L1 nem domináns az L2-vel szemben, és az L2 sem az L1-gyel szemben). Egy SL (L1) csak akkor nem összehasonlítható a másikkal (L2), ha a következő feltételek teljesülnek:
- Az L1 szakaszainak halmaza nem tartalmazza teljes egészében az L2 szakaszainak halmazát, és az L2 nem teljesen tartalmazza a L1 szakaszait. Ezáltal az L1 és L2 nem összehasonlítható.

Ha például egy L1 címkével rendelkező fájl szigorúan bizalmas információkat tartalmaz A és B szakasszal (TS A B), és az L2 minősített információkat tartalmazó fájl címkéje C szakasszal (C C), akkor az L1 nem összehasonlítható az L2-vel.

36. táblázat: Nem összehasonlítható SL-ek

L1		L2		Dominancia
Címke	**Szakasz**	**Címke**	**Szakasz**	
TOP SECRET	A, B	CLASSIFIED	C	-

Integritás címkék (TL):

A TL címkék jelölik a rendszerobjektumok vagy -folyamatok bizalmi szintjét. A TL címkék struktúrája az SL címkékkel azonos, azzal a kivétellel, hogy a TL címkék csak hierarchikus besorolással rendelkeznek és szakaszokkal nem.

A folyamatok egy objektumot csak akkor módosíthatnak vagy törölhetnek, ha a folyamat TL címkéje dominál az objektum TL címkéjével szemben. A folyamat egy objektumot csak akkor törölhet vagy nevezhet át, ha a folyamat TL címkéje dominál mind az objektum, mind pedig az objektumot tartalmazó könyvtár TL címkéjével szemben. A folyamat egy objektumhoz csak akkor férhet hozzá, ha az objektum TL címkéje dominál a folyamat TL címkéjével szemben.

Egy objektum vagy folyamat TL címkéje a **lstxattr** parancs segítségével határozható meg. Egy objektum vagy folyamat TL címkéje a **settxattr** parancs segítségével módosítható.

Alanyok és objektumok címkéi:

A Trusted AIX a folyamatokat alanyként azonosítja. Minden folyamat rendelkezik SL címkével.

A MAC ellenőrzések során használt SL a tényleges SL (ESL). Az ESL címkéknek a folyamat engedélytartományába kell esniük. Az engedélytartomány alulról és felülről zárt. A felső határ a maximális engedély (Max CL), az alsó határ a minimális engedély (Min CL). Az ESL, a Max CL és a Min CL a folyamat hitelesítési adatstruktúrájában kerül tárolásra, hozzárendelésükre a folyamat létrehozása során kerül sor. A Max CL címkének dominálnia kell a Min CL és az ESL címkével szemben, illetve az ESL címkének dominálnia kell a Min CL címkével szemben. A folyamat SL címkéinek beállítására, illetve listázására a **settxattr** és **lstxattr** parancsok használhatók.

A rendszer különböző objektumainak elérését szabályozni kell. Az objektum a következők bármelyike lehet:

- folyamat
- fájlok (adatfájlok vagy bináris fájlok)
- IPC objektumok, hálózati csomagok, stb.

Az MLS rendszer minden objektuma és alanya címkével rendelkezik.

Könyvtár
A könyvtárakhoz SL tartomány van rendelve; minimális SL és maximális SL. A maximális SL-nek dominálnia kell a minimális SL-lel szemben, vagy egyenlő lehet vele. A könyvtárban lévő minden fájl ebbe a tartományba esik.

Fájlok A szokásos fájlokhoz két SL van rendelve, de ezek értéke mindig ugyanaz. Azaz valójában csak SL-lel rendelkeznek. A szimbolikus hivatkozások különböző értékekkel rendelkezhetnek az SL-ekhez.

Speciális fájlok
A speciális fájlokhoz - mint például az eszközök, tty-k és fifo-k - maximális és minimális SL van rendelve. A könyvtár, fájlok és speciális fájlok csak egy integritási címkével (TL) rendelkeznek, a folyamatokhoz azonban egy maximális és minimális TL van rendelve.

Folyamat
Minden folyamathoz maximális és minimális érzékenységi engedélytartomány, valamint maximális és minimális integritási engedélytartomány van rendelve. Ezek az értékek a felhasználó engedély értékeiből öröklődnek. Az érzékenységi és integritási szint, amelyet a folyamat használ, a hatályos érzékenységi és integritási szint.

Felhasználói engedély címkék:

A felhasználók maximális és minimális érzékenységi engedély címkével (SCL), illetve maximális és minimális integritás engedély címkével (TCL) rendelkeznek.

Maximális és minimális érzékenységi engedély címkék

Minden felhasználó rendelkezik egy maximális érzékenységi engedély címkével (max SCL). A felhasználó max SCL címkéjének a felhasználó tényleges SL címkéjével szemben dominálnia kell. A max SCL segítségével bizonyos felhasználókra vonatkozóan korlátozható az érzékeny adatok megtekintése. A min SCL segítségével megakadályozható, hogy a magas biztonsági szinttel rendelkező felhasználók adatokat adjanak át az alacsonyabb biztonsági szinttel rendelkező felhasználóknak.

Tegyük fel például, hogy az A felhasználó max SCL és min SCL címkéje egyaránt PUBLIC_A, a B felhasználó max SCL és min SCL címkéje pedig egyaránt PUBLIC_B. Min SCL nélkül az A felhasználó információkat cserélhetne a B felhasználóval akkor, ha az IMPL_LO tényleges SL használatával bejelentkezne, majd az adatokat egy fájlba írná, amelyet a B felhasználó ezt követően el tudna olvasni. A min SCL használatával azonban az A felhasználónak PUBLIC_A címkével kell bejelentkeznie, és fájlokat csak a PUBLIC_A címkén írhat. A PUBLIC_A címkén írt fájlok a B felhasználó által nem olvashatók.

Maximális és minimális integritás engedély címkék

Minden felhasználó - az előzőkön kívül - rendelkezik egy maximális integritás engedély címkével (max TCL) is. A felhasználó max TCL címkéjének a felhasználó tényleges TL címkéjével szemben dominálnia kell. A max TCL segítségével bizonyos felhasználókra vonatkozóan korlátozható az érzékeny adatok megtekintése. A min TCL segítségével ezen kívül az is megakadályozható, hogy a magas biztonsági szinttel rendelkező felhasználók adatokat adjanak át az alacsonyabb biztonsági szinttel rendelkező felhasználóknak.

Fájlrendszer-objektumok címkéi:

Minden fájl konkrét biztonsági információkat tartalmaz. Új fájl létrehozásakor a fájl SL címkéje megegyezik a fájlt létrehozó folyamat SL címkéjével. A fájlban található információk SL címkéje kiemelése vagy visszaléptetése a fájl SL címkéjének növelésével vagy csökkentésével végezhető el.

A könyvtárakhoz létrehozásukkor minimális és maximális SL kerül hozzárendelésre. A létrehozáskor ezek úgy kerülnek beállításra, hogy mindkettő egyenlő legyen a létrehozó folyamat tényleges SL címkéjével, tehát valójában egy egyszintes könyvtár kerül létrehozásra. Ezeket az SL címkéket csak a megfelelő jogosultsággal és felhatalmazással rendelkező felhasználók módosíthatják. A könyvtárban új objektumok csak akkor hozhatók létre, ha az új objektumot létrehozó folyamat tényleges SL címkéje a könyvtár SL-tartományába esik.

Az ablakok rendszerint önálló leszármazott folyamatként kerülnek létrehozásra, és SL címkéjük egyenlő a felhasználó tényleges SL címkéjével. Az eszközökhöz (például az ablakokhoz tartozó pszeudó-terminálokhoz) szintén tartozik SL. A nevesített csővezetékek, amelyek a folyamatok közötti kommunikációra szolgáló eszközök, öröklik a nevesített csővezetéket létrehozó folyamat tényleges SL címkéjét. Az adatfolyamok, amelyek a folyamatok közötti kommunikációhoz kétirányú adatcsatornát biztosítanak, szintén öröklik az adatfolyamot létrehozó folyamat tényleges SL címkéjét.

Minden eszköz minimális és maximális SL címkével rendelkezik. A maximális SL címkének dominálnia kell a minimális SL címkével szemben. Alapértelmezésben a minimális és a maximális SL nem egyenlő. A folyamatok olvasási módban eszközhöz csak akkor férhetnek hozzá, ha a folyamat SL címkéje dominál az eszköz vagy könyvtár minimális SL címkéjével szemben. A folyamatok írási módban eszközhöz csak akkor férhetnek hozzá, ha a folyamat SL címkéje az eszköz vagy könyvtár minimális és maximális SL címkéje által meghatározott tartományra esik.

Fájlbiztonsági kapcsolók

Az objektumok megjelölhetők fájlbiztonsági kapcsolókkal (FSF), amelyek befolyásolják, hogy a folyamatok az objektumokat milyen módon kezelik. Az egyes FSF kapcsolók listáját, illetve a kapcsolók beállításához szükséges jogosultságok leírását a Fájlbiztonsági kapcsolók témakör tartalmazza. A folyamatok fájlbiztonsági kapcsolóval nem rendelkeznek.

Fájlok eltávolítása:

A fájlrendszerből objektumok csak az alábbi feltételek teljesülése esetén távolíthatók el:

- Az objektum eltávolítását megkísérlő folyamatnak látnia kell a fájl nevét a fájlt tartalmazó könyvtárban. Más szóval a folyamatnak keresési hozzáféréssel kell rendelkeznie az útvonal minden könyvtárban, egészen az eltávolítani kívánt objektumot tartalmazó könyvtárig, illetve a folyamat tényleges SL címkéjének dominálnia kell ezen könyvtárak mindegyikét. A fájlnév az **ls** parancs segítségével jeleníthető meg.
- A folyamatnak az eltávolítani kívánt objektumot tartalmazó könyvtárra vonatkozóan írási hozzáféréssel kell rendelkeznie.

Fájlok nyomtatása:

A nyomtatási alrendszer minden kimenetet automatikusan ellát a megfelelő érzékenységi címkékkel. Minden nyomtatási feladat automatikusan kap egy fejléc- és egy befejező oldalt, amely tartalmazza az összes biztonsággal kapcsolatos címkét és jelölést.

Fájlok mentése és visszaállítása:

Ha az AIX rendszeren lévő lemezekre vagy szalagokra ír a **backup** paranccsal, akkor az adatokhoz SL-ek kerülnek megadásra.

SO felhatalmazás szükséges a **backup** vagy **restore** parancsok használatához nem címkézett adatok szalagokról és lemezekről való importálásához vagy exportálásához. Nem címkézett adatok írásakor az adatok a SYSTEM_LOW alapértelmezett SL-t kapják fájlok, illetve a SYSTEM_LOW - SYSTEM_HIGH SL tartományt a könyvtárak esetén.

IPC objektumok címkéi:

Minden AIX IPC szolgáltatás magában foglalja a köztes objektumok létrehozását, illetve a köztes objektumok hozzáférését.

Az AIX három különböző IPC szolgáltatás meghatározását tartalmazza:

- Üzenetsorok
- Szemaforok
- Osztott memória

Mindhárom szolgáltatás magában foglalja a köztes (ún. IPC) objektumok létrehozását és hozzáférését a folyamatok közötti kommunikációhoz. Minden IPC objektumot egy attribútumhalmaz véd, hasonlóan a fájlokat védő attribútumokhoz. Az attribútumok:

- Az objektum tulajdonosának felhasználói és csoportazonosítója
- Az objektum létrehozójának felhasználói és csoportazonosítója
- Az erőforrás hozzáférési módja, amely hasonló a fájlhozzáférés esetében megismert jogosultsági bitekhez. Minden objektum rendelkezik olvasási, írási és végrehajtási hozzáféréssel az objektum tulajdonosára, a csoportra, illetve mindenki másra vonatkozóan.
- Az erőforrás-használat nyomon követéséhez használt sorszám
- Az erőforrást azonosító kulcs

Hasonlóan a többi rendszerobjektumhoz, a Trusted AIX ezeket az attribútumokat további biztonsági attribútumokkal terjeszti ki. A Trusted AIX rendszereken az IPC objektumok - az előzők mellett - az alábbi attribútumokkal is rendelkeznek:

- Érzékenységi címke (SL)
- Integritás címke (TL)

Az IPC objektumok biztonsági attribútumait a **settxattr** parancs segítségével jelenítheti meg. Az IPC objektum attribútumainak olvasása az objektumra vonatkozóan DAC READ és MAC READ hozzáférést igényel.

IPC objektumok elérése:

IPC objektumok számos rendszerhíváson keresztül kerülnek létrehozásra, törlésre és elérésre; ezek leírása a Trusted AIX programozás témakörben található. Normál felhasználók ezeket a műveleteket nem hajtják végre. A témakör az IPC objektumok létrehozására, törlésére és elérésére vonatkozó szabályok általános áttekintését tartalmazza.

Az IPC objektumok eléréséhez a folyamatoknak DAC, MIC és MAC hozzáférés-ellenőrzésen kell átesniük.

A DAC hozzáférés-ellenőrzések alapjául az objektum módja (tulajdonos, csoport vagy mindenki más) és a folyamat felhasználói és csoportazonosítói szolgálnak. A folyamatok akkor rendelkeznek DAC tulajdonosi hozzáféréssel egy IPC objektumhoz, ha a folyamat hatályos UID azonosítója megegyezik az objektum tulajdonosának vagy létrehozójának UID azonosítójával. Ugyanez igaz a DAC csoporthozzáférésekre is.

A MAC hozzáférés a folyamat és az objektumok SL címkéi alapján történik. A MIC hozzáférés a folyamat és az objektumok TL címkéi alapján történik.

Az IPC objektumok tartalmának hozzáférési szabályai megegyeznek az IPC objektumattribútumok hozzáférési szabályaival. Az IPC objektumok tartalmának vagy attribútumainak olvasásához DAC READ, MIC READ és MAC READ hozzáférés szükséges. Az IPC objektumok írásához DAC WRITE, MIC WRITE és MAC WRITE hozzáférés szükséges.

Az IPC objektumattribútumok az IPC objektumtartalomnál szigorúbban korlátozottak. Az IPC objektumattribútumok módosítása - ennek megfelelően - magasabb szintű jogosultságot igényel. Az általános AIX attribútumok (például: mód) módosításához a folyamatnak az objektumra vonatkozóan DAC OWNER és MAC WRITE hozzáféréssel kell rendelkeznie. Az IPC objektum SL címkéjének módosításához a folyamatnak az összes alábbiakban felsorolt jogosultsággal rendelkeznie kell:

- PV_SL_PROC jogosultság
- DAC OWNER (csak visszaléptetés)
- DAC WRITE
- MAC WRITE
- PV_SL_UG jogosultság az SL kiemeléséhez, illetve PV_SL_DG jogosultság az SL visszaléptetéséhez
- PV_MAC_CL, ha a létező vagy új SL a folyamat jogosultságán kívüli
- MIC WRITE

Az IPC objektum TL címkéjének módosításához a folyamatnak az összes alábbiakban felsorolt jogosultsággal rendelkeznie kell:

- PV_TL jogosultság
- DAC OWNER
- MAC WRITE
- MIC WRITE

Ezen kívül a memória osztott szegmensének zárolásához, illetve a zárolás feloldásához a folyamatnak a PV_KER_IPC_O jogosultsággal is rendelkeznie kell. Ha egy folyamat az $_{msgctl}$ szubrutinban egy üzenetsor msg qbytes paraméterét módosítja, akkor PV_KER_IPC jogosultsággal is rendelkeznie kell.

Kapcsolódó fogalmak:

“Trusted AIX programozása” oldalszám: 445

A rendszerbiztonság a Megbízható számítástechnikai alapkörnyezet (TCB) szoftverétől, hardverétől és firmware-jétől függ. Ez a teljes operációs rendszer kernelt, minden eszközillesztőt és System V STREAMS modult, kernelbővítményt és minden megbízható programot tartalmaz. A programok által használt fájlokat a rendszer a biztonsági döntésekben a TCB részének tekinti.

IPC objektum létrehozása és törlése:

Az IPC objektum létrehozására vonatkozóan nem léteznek megszorítások. Amikor egy folyamat IPC objektumot hoz létre, akkor az objektum örökli a folyamat SL és TL címkéjét.

Az IPC objektum hozzáférési módját az objektumot létrehozó rendszerhívásnak kell meghatároznia.

IPC objektumok törléséhez a folyamatnak az objektumra vonatkozóan DAC OWNER, MIC WRITE és MAC WRITE hozzáféréssel kell rendelkeznie.

Trusted Networking:

Biztonságosi hálózatkezelési követelmények szükségesek a továbbfejlesztett biztonságosi rendszerek kiterjesztett biztonsági attribútumaihoz. Az AIX Trusted Network számos ismert biztonságos hálózatkezelési szabványt támogat, az U.S. DoD RFC1108 Revised Internet Protocol Security Option (RIPSO) és Commercial Internet Protocol Security Option (CIPSO) szabványt is beleértve.

Az AIX Trusted Network az IPv4 és IPv6 protokollt egyaránt támogatja. Más megbízható rendszerekkel való kommunikációnál az SL befoglalásra kerül az IP elemekbe a CIPSO/RIPSO szabványoknak megfelelően. Az IP rétegben a csomagokon küldött SL-ekhez kikényszerítésre kerülnek a MAC ellenőrzések. Az engedélyezett címketartomány hálózati szabályokkal kerül beállításra. A hálózati szabályok hosztszabályokból és csatolószabályokból állnak. Az AIX Trusted Network csak az alapértelmezett csatolószabályokat telepíti (beállított csatolónként egy szabály). Beállíthat hosztszabályokat finomabb szűrés lehetővé tétele érdekében. A netrule parancs segítségével hoszt- és csatolószabályokat is beállíthat. A netrule parancs által támogatott műveletek: szabályok hozzáadása, törlése, listázása és lekérdezése.

A tninit parancs segítségével inicializálhatja a Trusted Network alrendszert és karbantarthatja a Trusted Network szabályadatbázist.

Root letiltása:

A Trusted AIX rendszereken a root felhasználói fiók tiltott. Ennek köszönhetően minimálisra csökkenthetők az olyan károk, amelyeket a rendszeren egy minden jogosultsággal rendelkező felhasználó okozhat.

A root felhasználóként történő bejelentkezés minden típusa tiltott. A root felhasználói bejelentkezéseket csak a **su** parancs teszi lehetővé. A root által birtokolt folyamatokhoz különleges jogosultságok nem kerülnek hozzárendelésre. A root által birtokolt setuid és nem setuid programok az előzőek során megszokott módon működnek akkor, ha azokat felhatalmazott felhasználók hívják meg. A felhatalmazással nem rendelkező felhasználók esetében a program akkor futtatható, ha a DAC módbitek vagy ACL listák a végrehajtást engedélyezik, de a programhoz jogosultságok nem kerülnek hozzárendelésre. Ennek következtében előfordulhat, hogy a program privilegizált műveleteket nem tud végrehajtani akkor, ha felhatalmazással nem rendelkező felhasználók futtatják. Tehát az újonnan telepített alkalmazásokhoz megfelelő jogosultságokat kell hozzárendelni akkor, ha az alkalmazásoknak privilegizált műveleteket kell végezniük.

A rendszer adminisztrációs feladatait az információrendszer adatvédelmi megbízott (ISSO), rendszeradminisztrátor (SA), illetve rendszerfelelős (SO) szereppel rendelkező felhasználók végezhetik el. A szerepek segítségével tetszőleges felhasználó végrehajthat rendszeradminisztrációs feladatokat.

Megjegyzés: A Trusted AIX telepítése során a root fiók **su** attribútumát a rendszer false értékre állítja. Ahhoz, hogy az egyéb adminisztrátori felhasználók a root fiókhoz hozzá tudjanak férni, az ISSO felhatalmazással rendelkező felhasználónak az attribútum értékét a **chuser** parancs segítségével vissza kell állítania true értékre, majd a fiókhoz jelszót kell rendelnie.

Címketámogatás a megfigyelés során:

A megfigyelési alrendszer elsődleges célja a biztonsággal kapcsolatos események megfigyelése és rögzítése.

A megfigyelési alrendszer által biztosított információk segítségével az alábbi típusú információk rögzíthetők:
- A biztonsági irányelv megsértésére tett kísérletek
- A biztonsággal kapcsolatos tevékenység sikeres befejezése

A megfigyelési alrendszer az alábbi képességeket biztosítja:
- A megfigyelendő események körének meghatározása
- A megfigyelés ki- és bekapcsolása a rendszer futása során
- A megfigyelési naplófájlok közötti zökkenőmentes (információvesztés nélküli) váltás
- A megfigyelési információk felhasználó által olvasható formátumúra alakítása
- A megfigyelési információk részhalmazának kiválasztása és feldolgozása

A megfigyelési alrendszer beállításakor az ISSO felhasználónak ismernie kell, hogy mit kell megfigyelni, a megfigyelés milyen helyzetekben következik be, illetve a megfigyelés milyen módon kezdeményezhető és állítható le. A megfigyelés beállításával, indításával és leállításával, felügyeletével és áttekintésével kapcsolatosan részletes információkat a Megfigyelés áttekintése témakör tartalmaz.

A megfigyelési alrendszer kikapcsolás, rendszerösszeomlás, áramkimaradás és egyéb problémák esetén állapotát megtartja, illetve automatikusan újraindul. A megfigyelési alrendszer képes saját magát automatikusan lezárni, a rendszert leállítani, illetve a meglévő megfigyelési fájlról másikra váltani akkor, ha olyan helyzet áll elő, hogy a megfigyelési rekordokat a meglévő megfigyelési fájlban már nem tudja tárolni. A megfigyelési fájlok között automatikusan lehet váltani akkor is, ha a fájlrendszer telítettsége egy megadott szintet elér. Azonban súlyos áramkimaradás esetén lehetséges, hogy a megfigyelési rekordok egy része elvész.

Többszintű és particionált könyvtárak:

A többszintű könyvtárak olyan általános könyvtárak, amelyekhez egyetlen SL helyett SL-tartomány kerül hozzárendelésre. A particionált könyvtárak a felhasználó számára egyetlen könyvtárként jelennek meg. Azonban a felhasználónak megjelenő fájlok valójában a particionált könyvtár egyik rejtett alkönyvtárában találhatók.

Többszintű könyvtárak:

A többszintű könyvtárak olyan általános könyvtárak, amelyekhez egyetlen SL helyett SL-tartomány kerül hozzárendelésre.

A többszintű könyvtárakban található fájlok neveinek megjelenítéséhez a folyamatnak a könyvtár minimális SL címkéjénél magasabb biztonsági szinten kell működnie. A tényleges fájlok létrehozásához vagy törléséhez a folyamatnak a többszintű könyvtár SL-tartományán belül kell működnie.

A többszintű könyvtárakban található minden fájl saját SL címkével rendelkezik, és minden fájlt a szabványos MAC megszorítások védenek. Azonban az adott könyvtárhoz hozzáféréssel rendelkező összes folyamat képes a könyvtárban található összes objektum nevének megjelenítésére. Ennek megfelelően lehetséges, hogy egy folyamat egy adott könyvtárban MAC írási és olvasási képességekkel rendelkezik, de mégsem tud a könyvtárban bizonyos fájlokat írni és/vagy olvasni, bár a folyamat képes a könyvtárban található összes fájl nevének megjelenítésére.

Particionált könyvtárak:

A particionált könyvtárak a felhasználó számára egyetlen könyvtárként jelennek meg. Azonban a felhasználónak megjelenő fájlok valójában a particionált könyvtár egyik rejtett alkönyvtárában találhatók.

A többszintű könyvtárak biztonságkockázati tényezőt jelentenek. A magas biztonsági szinten működő folyamatok beolvashatnak egy alacsonyabb biztonsági szintű fájlt, majd fájlokat hozhatnak létre ugyanazon a magas biztonsági szinten. Noha a MAC szolgáltatások megakadályozzák, hogy az alacsonyabb biztonságú folyamatok az új fájlokat beolvassák, az alacsonyabb biztonságú folyamatok ettől függetlenül látják az új fájlok nevét. Ha a magas biztonságú folyamat az új fájlok nevét az eredeti magas biztonságú fájlok alapján határozza meg, akkor az alacsonyabb biztonsági szinttel rendelkező folyamatok hozzáférést nyerhetnek a magasabb biztonsági szintű információkhoz, egyszerűen az új fájlnevek beolvasásával.

Ha egy particionált könyvtárra létrehozása után egy folyamat hivatkozik, akkor a rendszer létrehoz egy rejtett alkönyvtárat, amelynek SL címkéje megegyezik a hivatkozó folyamatéval. Ha a folyamat ezt követően létrehoz egy fájlt, akkor a fájl ténylegesen a rejtett alkönyvtárban kerül létrehozásra. A particionált könyvtárak számos ilyen rejtett alkönyvtárat tartalmazhatnak, de a particionált könyvtárra hivatkozó folyamat csak a hivatkozó folyamattal egyező SL címkével rendelkező rejtett alkönyvtárban található fájlokat látja. Amikor egy folyamat particionált alkönyvtár alkönyvtárát hozza létre, akkor az alkönyvtár particionált alkönyvtár lesz.

A particionált könyvtárhoz SYSTEM_LOW - SYSTEM_HIGH SL tartomány van rendelve. Ezáltal minden folyamat elérheti a particionált könyvtárakat.

Az **aix.mls.pdir.mkdir** felhatalmazással rendelkező felhasználók a **pdmkdir** parancs segítségével hozhatnak létre particionált könyvtárakat. Az üres particionált könyvtárak a **pdrmdir** parancs segítségével távolíthatók el. A normál könyvtárak a **pdset** parancs segítségével módosíthatók particionált típusúra. A particionált könyvtárak normál típusúra módosítására parancs nem létezik.

A particionált könyvtárakon belül az egyik particionált alkönyvtárban található fájl csatolható az ugyanabban a particionált könyvtárban található összes magasabb SL címkével rendelkező particionált alkönyvtárhoz. Ennek köszönhetően egy adott fájlhoz az adott particionált könyvtárhoz, illetve az adott particionált könyvtár magasabb szintű particionált alkönyvtáraihoz hozzáféréssel rendelkező összes folyamat hozzáférhet. Az ilyen típusú fájlcsatolás a **pdlink** parancs segítségével végezhető el.

Particionált könyvtárak hozzáférési módjai:

A folyamatokhoz létrehozásukkor hozzárendelésre kerül a két lehetséges mód (valós mód vagy virtuális mód) egyike. A mód meghatározza, hogy a folyamat a particionált könyvtárakat milyen módon jeleníti meg.

A valós módú folyamatok a particionált könyvtárakat szabványos többszintű könyvtárként kezelik. A megszokott DAC, MIC és MAC megszorítások kielégítésével minden particionált alkönyvtár elérhető szabványos könyvtárként. A valós módú folyamatok beléphetnek egy particionált könyvtárba, illetve - a DAC, MIC és MAC megszorítások figyelembevételével - megjeleníthetik a könyvtár összes alkönyvtárát.

A virtuális módú folyamatok sosem lépnek be a particionált könyvtárakba; ehelyett átirányításra kerülnek abba a particionált alkönyvtárba, amelynek maximális és minimális SL címkéi egyaránt egyenlők a folyamat tényleges SL címkéjével.

A valós módú folyamatok a parancsokat virtuális módban a **pdmode** parancs segítségével futtathatják (például: pdmode ls). Hasonlóképpen a virtuális módú folyamatok is futtathatnak parancsokat valós módban, szintén a **pdmode** parancs segítségével (például: pdmode -r ls). Azonban ez aix.mls.pdir.mode felhatalmazást igényel. A felhatalmazás birtokában a felhasználó virtuális módban futó parancsértelmezőről valós módban futó parancsértelmezőre a pdmode -r sh parancs futtatásával válthat át. Ha valós módban futás esetén egy programot virtuális módban kíván indítani, akkor ahhoz felhatalmazás nem szükséges.

Könyvtártípusok megjelenítése és módosítása:

Az **lstxattr** parancs segítségével a könyvtártípust a secflags attribútum részeként jelenítheti meg. Az FSF_PDIR particionált könyvtárat, az FSF_PSDIR particionált alkönyvtárat, az FSF_PSSDIR pedig particionált al-alkönyvtárat jelöl. A **pdset** paranccsal a normál könyvtártípust particionált könyvtártípusra cserélheti.

Trusted AIX adminisztráció

Trusted AIX rendszer kezelése számos Trusted AIX rendszerre jellemző tényezőt foglal magában.

Trusted AIX telepítése

A Trusted AIX csak az alapszintű operációsrendszer-telepítés során, a telepítési menü Biztonsági modell lehetőségével engedélyezhető.

A Trusted AIX átállítási lehetősége nem támogatott. A megtartási telepítéshez JFS2 fájlrendszereket kell használni. A felhasználói interakció nélküli hálózati telepítés esetében az alapértelmezett adminisztrátori felhasználókhoz tartozó jelszavakat a 37. táblázat: témakör tartalmazza.

37. táblázat: Alapértelmezett adminisztrátori felhasználók jelszavai

Felhasználó	Jelszó
isso	isso
sa	sa
so	so

Futtatási módok

A rendszeren két futtatási mód (a konfigurációs mód és a működési mód) áll rendelkezésre a rendszerkonfiguráció és karbantartás, illetve a mindennapos üzemeltetés céljából.

Rendszerbetöltéskor a rendszer kezdetben konfigurációs módban fut. Az inicializálás befejezésével a futtatási mód működési módra vált.

A rendszer karbantartása és helyreállítása a konfigurációs mód segítségével végezhető el. Amikor a rendszer egyfelhasználós módban kerül betöltésre, akkor a rendszer minimálisan beállított, illetve a hálózatkezelés tiltott. A konfigurációs mód a rendszer kritikus, biztonsággal kapcsolatos részeinek adminisztrációjára használható.

A működési mód az általános rendszerműködési mód. A rendszer akkor vált át erre a módra, ha az összes olyan feladat, amelynek be kell lépnie az alapértelmezett futtatási szintre, már befejeződött.

A rendszer futtatási módja a **getrunmode** parancs segítségével jeleníthető meg, illetve a **setrunmode** parancs segítségével módosítható.

Kernelbiztonsági kapcsolók

A kernelbiztonsági kapcsolók segítségével bizonyos biztonsági összetevők engedélyezhetők és tilthatók le, például a címkeellenőrzés foganatosítása, az integritás címkék ellenőrzése az olvasási műveletek során stb.

A biztonsági ellenőrzések foganatosítása előtt a kernel ellenőrzi a kernelbiztonsági kapcsolókat. A kapcsolók használata csak akkor támogatott, ha a Trusted AIX engedélyezett. A felhasználói tárterületen a kapcsolókat az ODM adatbázis tárolja. A rendszer futási módjának függvényében a kernel a megfelelő kernelbiztonsági kapcsolókat ellenőrzi.

38. táblázat: Kernelbiztonsági kapcsolók és alapértelmezett értékeik

Kernelbiztonsági kapcsoló	Engedélyezett	Tiltott	Működési mód alapértelmezése	Konfigurációs mód alapértelmezése
tnet_enabled	A Trusted Network funkcionalitás elérhető	A Trusted Network funkcionalitás nem állítható be és nem használható	Tiltott	Tiltott
tl_write_enforced	Írási, törlési és átnevezési művelet esetében a MIC foganatosítja	A beállítások szerint a TL címkéket a rendszer az írási ellenőrzések során nem használja	Engedélyezett	Engedélyezett
tl_read_enforced	Olvasási művelet esetében a MIC foganatosítja	Beállítások szerint a TL címkéket a rendszer az olvasási ellenőrzések során nem használja	Tiltott	Tiltott
sl_enforced	MAC foganatosítja	A beállítások szerint az SL címkéket a rendszer a hozzáférés-felügyelet során nem használja	Engedélyezett	Tiltott
trustedlib_enabled	A fájlrendszer-objektumok FSF_TLIB kapcsolóját a rendszer tiszteletben tartja	Az FSF_TLIB kapcsolókat a rendszer nem tartja tiszteletben	Tiltott	Tiltott

Kernelparaméterek beállítása

A Trusted AIX kernel beállítható úgy, hogy a telephelyen érvényes biztonsági irányelvek által megkövetelt biztonsági megszorításokat foganatosítsa.

A biztonsági beállítások a **getsecconf** parancs segítségével jeleníthetők meg, illetve a **setsecconf** parancs segítségével módosíthatók. A beállítható kernelparaméterek:

- Érzékenységi címkék foganatosítása
- Integritásolvasás foganatosítása
- Integritásírás foganatosítása
- Trusted Network
- Megbízható függvénytár

A paraméterek csak a rendszer konfigurációs futtatási módjában állíthatók be.

/etc/security/enc/LabelEncodings fájl személyre szabása

A rendszer címkéinek meghatározását az /etc/security/enc/LabelEncodings fájl tartalmazza, amely minden telephelyre személyre szabható.

A címkék a Trusted AIX telepítését követően szabhatók személyre.

A Trusted AIX rendszerek rendelkeznek egy meghatározott SYSTEM LOW SL (SLSL) címkével, amelyet a rendszer összes egyéb érzékenységi címkéje dominál, illetve egy SYSTEM HIGH SL (SHSL) címkével, amely az összes többi érzékenységi címkét dominálja. Hasonlóképpen, a SYSTEM LOW TL (SLTL) címkét a rendszer összes egyéb integritás címkéje dominálja, míg a SYSTEM HIGH TL (SHTL) az összes egyéb integritás címkét dominálja. A meghatározások a legmagasabb és legalacsonyabb SL és TL címkék értékeit veszik fel, az /etc/security/enc/LabelEncodings fájlban meghatározott módon.

A Trusted AIX rendszerek betöltésekor az /etc/security/enc/LabelEncodings fájlban található rendszercímkék betöltésre kerülnek a kernelbe. A kernelbe a címkék ezen kívül a **setsyslab** parancs segítségével is betölthetők. A kernelben meghatározott rendszercímkék listája a **getsyslab** parancs segítségével jeleníthető meg. Az /etc/security/enc/LabelEncodings fájl módosítását követően ajánlott a rendszert újraindítani.

Az /etc/security/enc/LabelEncodings fájlba megjegyzések tetszőleges kulcsszó kezdeténél elhelyezhetők. A megjegyzések * karakterrel kezdődnek, és egészen a sor végéig tartanak.

Az /etc/security/enc/LabelEncodings fájl a változati információkat, illetve az alábbi kötelező szakaszokat tartalmazza. Minden szakasznak az alábbi szakaszkulcsszavak valamelyikével és egy kettősponttal (:) kell kezdődnie.

- classifications
- information labels
- sensitivity labels
- clearances
- channels
- printer banners
- accreditation range

Az /etc/security/enc/LabelEncodings fájl a VERSION bejegyzéssel kezdődik. A bejegyzés egy karaktersorozat, amely tartalmazhat szóközszerű karaktereket.

Az alábbi kulcsszavak közül az egyes szakaszokban mindegyik megjelenhet. A kulcsszavak lezárása pontosvesszővel (;) történik:

name=*név*
A besorolás vagy szakasz teljes nevének meghatározásához használható kulcsszó.

sname=*név*
A rövidített név meghatározásához használható kulcsszó. Nem kötelező.

aname=*név*
Alternatív besorolási kulcsszó. Nem kötelező.

value=*érték*
A besorolás vagy szakasz belső egész szám értékének meghatározásához használható kulcsszó.

compartments=*bit*
A kulcsszó meghatározza, hogy melyik szakaszbit legyen 0 vagy 1, ha a szót a címke tartalmazza

Címkekódolási formátum Trusted AIX bővítései

A Védelmi Hírszerzési Ügynökség dokumentuma (DDS-2600-6216-93) által előírt címkekódolás az integritás címkéket nem támogatja.

Alapértelmezésben az érzékenységi címkéket integritás címkékként használjuk. A Trusted AIX támogatja a nem kötelező integritás címkék szakasz használatát, amely az érzékenységi címkék szakasztól eltérő lehet. Ennek a rugalmasságnak köszönhetően az érzékenységi és integritás címkék eltérő besorolással rendelkezhetnek. Az érzékenységi címkék például elláthatók SL előtaggal, az integritás címkék pedig TL előtaggal, az alábbiak szerint:

39. táblázat: Érzékenységi címkék besorolási nevei és értékei

name	sname	value
name= SL IMPLEMENTATION LOW	sname= SL_IMPL_LO	value= 0
name= SL UNCLASSIFIED	sname= SL_U	value= 20
name= SL PUBLIC	sname= SL_PUB	value= 40
name= SL SENSITIVE	sname= SL_SEN	value= 60
name= SL RESTRICTED	sname= SL_RES	value= 80
name= SL CONFIDENTIAL	sname= SL_CON	value= 100
name= SL SECRET	sname= SL_SEC	value= 120
name= SL TOP SECRET	sname= SL_TS	value= 140

40. táblázat: Integritás címkék besorolási nevei és értékei

name	sname	value
name= TL IMPLEMENTATION LOW	sname= TL_IMPL_LO	value= 0
name= TL UNCLASSIFIED	sname= TL_U	value= 20
name= TL PUBLIC	sname= TL_PUB	value= 40
name= TL SENSITIVE	sname= TL_SEN	value= 60
name= TL RESTRICTED	sname= TL_RES	value= 80
name= TL CONFIDENTIAL	sname= TL_CON	value= 100
name= TL SECRET	sname= TL_SEC	value= 120
name= TL TOP SECRET	sname= TL_TS	value= 140

Az integritás címke szakaszra az alábbi szabályok érvényesek:

- Az "INTEGRITY LABELS" szakasz csak a "NAME INFORMATION LABELS" szakasz után vehető fel. Olyan esetekben, ahol az adminisztrátor nem adta meg az elhagyható "NAME INFORMATION LABELS" szakaszt, az "INTEGRITY LABELS" szakaszt az "ACCREDITATION RANGE" szakasz után kell megadni.
- A címkekódolási fájlban csak egy "INTEGRITY LABELS" szakasz lehet. Az objektumokra és alanyokra ugyanaz a szakasz vonatkozik.
- Az új "INTEGRITY LABELS" szakasz elhagyható. Ha a szakaszt elhagyja, akkor a kötelező "CLASSIFICATIONS" szakaszban megadott besorolásokat kell használni.
- Az "INTEGRITY LABELS" szakasznak a "CLASSIFICATIONS" szakaszhoz hasonlónak kell lennie. A következő kulcsszavakat tartalmazhatja: **"name="**, **"sname="**, **"aname="** és **"value="**. A "CLASSIFICATIONS" szakasz részét képező **"initial compartments="** és **"initial markings="** kulcsszó az "INTEGRITY LABELS" szakaszban nem érvényes.
- A **"value="** adattartománya megegyezik a "CLASSIFICATIONS" szakasz adattartományával (0 - 32 000).

Rendszer indítása

A rendszerbiztonság automatikusan meghívásra kerül a rendszerindítási szekvencia során. Ellenőrizni kell, hogy az indítási szekvencia során megjelenő biztonsági paraméterek megfelelők-e a rendszer számára.

Konfiguráció indítási mód:

A rendszer karbantartása és helyreállítása a konfigurációs mód segítségével végezhető el.

Amikor a rendszer egyfelhasználós módban kerül betöltésre, akkor a rendszer beállítása minimális, illetve a hálózatkezelés tiltott.

Működési indítási mód:

A napi működés során a rendszer a működési módot használja.

Normális esetben a rendszert ajánlott közvetlenül többfelhasználós módban betölteni. Ha a rendszerbetöltési felhatalmazási program érvényes felhasználónevet és jelszót kap, akkor a rendszer belép a működési módba, megjelenik a bejelentkezési hitelesítési képernyő, majd az érvényes felhasználók bejelentkezhetnek.

A biztonsági mechanizmusok (például az érzékenységi címkék, a tetszés szerinti hozzáférés-felügyelet, a MAC, a jogosultságellenőrzések, az azonosítás és hitelesítés, illetve a felhatalmazások) konfigurációs és működési módban egyaránt aktívak, a megfelelő biztonsági konfigurációs kapcsolók által meghatározott módon. További információkat a **getsecconf** parancs leírása tartalmaz.

Az összes elvárt rendszerfunkcionalitás biztosításához ajánlott a rendszert csak működési módban működtetni.

Rendszerbetöltési folyamat:

A Trusted AIX rendszereken az új rendszerbetöltő parancsfájlok az /etc/inittab fájlhoz kerülnek hozzáadásra. Az új rendszerbetöltő parancsfájlok az rc.mls.boot, az rc.mls.net, illetve az rc.mls. Ezek ebben a sorrendben kerülnek végrehajtásra.

Az rc.mls.boot parancsfájlban végrehajtott lépések:

1. Interaktív integritásellenőrzés kerül végrehajtása, amely a felhasználótól információkat kér az egyes eltérések kezelési módjára vonatkozóan (a **trustchk** parancs használatával)
2. Beállításra kerülnek a konfigurációs mód kernelbiztonsági kapcsolói (a **setsecconf** parancs használatával)
3. Beállításra kerülnek a rendszercímkék (minimális és maximális érzékenységi címkék, illetve integritás címkék)
4. A konfigurációs mód kernelbiztonsági kapcsolói megjelenítésre kerülnek a képernyőn

Az rc.mls.net parancsfájlban végrehajtott lépések:

1. Trusted AIX alrendszer inicializálása.
2. Ha az /etc/security/rules.int fájl létezik, akkor a szabályadatbázis betöltésre kerül a kernelbe.

Az rc.mls parancsfájlban végrehajtott lépések:

1. Trusted AIX alrendszer inicializálása.
2. Ha az /etc/security/rules.int fájl létezik, akkor a szabályadatbázis betöltésre kerül a kernelbe.

Megjegyzés: A rendszerbetöltő parancsfájlok bármilyen módosítása a rendszer meghibásodásához vezethet.

Rendszerindítás személyre szabása:

Ugyan nem ajánlott, a rendszer indításakor történő rendszerbetöltési hitelesítés és a rendszerintegritás ellenőrzése letiltható.

A rendszerkonzolnál egy operátornak jelen kell lennie a rendszer indításához, hacsak a rendszerbetöltési hitelesítés és a rendszerintegritás ellenőrzése nem tiltott.

BOOT hitelesítés letiltása:

A BOOT hitelesítés az **rmitab bootauth** parancs futtatásával vagy az SMIT menü segítségével tiltható le.

Rendszerintegritás-ellenőrzés letiltása:

Az automatikus rendszerbetöltési integritásellenőrzés letiltásához távolítsa el az **rc.mls.boot** parancsfájl trustchk sorát.

A rendszer leállítása

A rendszerleállítás privilegizált művelet és az aix.system.boot.shutdown felhatalmazás védi.

SO szereppel valamint ezt a felhatalmazást birtokló szereppel rendelkező felhasználók leállíthatják a rendszert.

Megbízható helyreállítás

Előfordulhat, hogy a rendszer nem tiszta állapotban áll le. Ennek oka lehet tápellátás-kimaradás, véletlen leállítás vagy hardverhiba. A Trusted AIX ezekből a körülményekből speciális újraindítási eljárások nélkül helyre tud állni.

A rendszer újraindításakor minden védelmi mechanizmus aktív a rendszer leállításai módjától függetlenül. A rendszerindítási eljárás során minden fájlrendszer automatikusan ellenőrzésre kerül a veszteségre vonatkozóan, mielőtt a felhasználók bejelentkeznének. Az indítási parancsfájlok futtathatják az **fsck** parancsot a biztonságossá tétel vagy annak megakadályozása érdekében, hogy a jogosulatlan felhasználók sérült vagy veszélyeztetett fájlokat érjenek el.

A **trustchk** parancs jelenti a fájlok vagy könyvtárak biztonsági attribútumainak inkonzisztenciáit, és interaktív módon felszólítja a felhasználót ezen attribútumok kijavítására. A **trustchk** parancsnak futnia kell, ha lehetőség van a fájlrendszer integritásának veszélyeztetésére. További információkért tekintse meg a **trustchk** parancsot.

Bejelentkezés

Ahhoz, hogy a Trusted AIX felhasználók a rendszerre bejelentkezhessenek, megfelelő érzékenységi és integritás engedélyeket kell hozzájuk rendelni.

A felhasználó engedélyeinek meghatározását - felhasználói attribútumok formájában - az /etc/security/user fájl tartalmazza. A minsl és maxsl attribútumok meghatározzák a felhasználó érzékenységi engedélyét. A mintl és maxtl attribútumok meghatározzák a felhasználó integritás engedélyét. A defsl és deftl attribútumok meghatározzák a felhasználó bejelentkezéskor érvényes tényleges érzékenységi és integritás szintjét.

A felhasználó engedélyattribútumai a chuser és chsec parancsok segítségével módosíthatók, az attribútumok listája a lsuser és lssec parancsok segítségével jeleníthető meg.

A felhasználók saját címkéiket listázhatják, de nem módosíthatják. Az egyéb felhasználók engedélyszintjeinek listázásához a felhasználónak aix.mls.clear.read felhatalmazással kell rendelkeznie. Az engedélyek módosításához a felhasználónak aix.mls.clear.write felhatalmazással kell rendelkeznie.

A bejelentkezéshez az összes alábbi szabálynak teljesülnie kell:

- A minsl értékkel szemben a defsl értéknek dominálnia kell
- A defsl értékkel szemben a maxsl értéknek dominálnia kell
- A mintl értékkel szemben a deftl értéknek dominálnia kell
- A deftl értékkel szemben a maxtl értéknek dominálnia kell

A kívánt tényleges érzékenységi és integritás szintek bejelentkezéskor a **login** parancs **–e** és **–t** paraméterével adhatók meg. További információkat a **login** parancs leírása tartalmaz.

Ha a rendszer akkreditációs tartományán kívül eső érzékenységi címkével kíván bejelentkezni, akkor aix.mls.label.outsideaccred felhatalmazással kell rendelkeznie.

A Trusted AIX a rendszerfelhasználók (128-nál kisebb felhasználói azonosítóval rendelkező felhasználók) bejelentkezését nem engedélyezi.

Bejelentkezési hibák okai

A bejelentkezésre tett kísérlet számos okból meghiúsulhat.

Ha az alábbi feltételek bármelyike teljesül, akkor a bejelentkezési kísérlet meghiúsul:

- A megadott bejelentkezési azonosító érvénytelen
- A megadott jelszó érvénytelen
- A fiók zároltként jelölt, mert a fiók esetében a hibás bejelentkezési kísérletek száma meghaladta a rendszeren meghatározott korlátot
- A port zároltként jelöl, mert a port esetében a hibás bejelentkezési kísérletek száma meghaladta a rendszeren meghatározott korlátot
- A bejelentkezési azonosító nem rendelkezik érvényes engedéllyel
- A megadott címke (vagy ha nincs megadott címke, akkor a bejelentkezési azonosító alapértelmezett érzékenységi vagy integritás címkéje) érvénytelen, a bejelentkezési azonosító engedélyén kívüli, a bejelentkezési eszköz engedélyén kívüli vagy a rendszer akkreditációs tartományán kívüli
- A felhasználó a bejelentkezési parancsértelmező program útvonalnevére vonatkozóan nem rendelkezik DAC hozzáféréssel, vagy a felhasználói fiók a bejelentkezési parancsértelmező programhoz nem rendelkezik DAC végrehajtási hozzáféréssel

- A felhasználó a bejelentkezési parancsértelmező program útvonalnevére vonatkozóan nem rendelkezik MAC vagy MIC hozzáféréssel vagy a bejelentkezési parancsértelmező programhoz nem rendelkezik MAC vagy MIC olvasási hozzáféréssel
- A bejelentkezési azonosító uid azonosítója 128-nál kisebb

Felhasználó átkapcsolása a su paranccsal

Trusted AIX rendszeren a **su** parancs **-** paraméterrel való meghívása esetén az aktuális felhasználó engedélyeinek dominálnia kell az új felhasználó engedélyszintjével szemben.

A következő feltételeknek kell teljesülnie az érzékenységi és integritási címkékhez:

- az aktuális felhasználó maximális engedélyének dominálnia kell az új felhasználó maximális engedélyével szemben.
- az új felhasználó minimális engedélyének dominálnia kell az aktuális felhasználó minimális engedélyével szemben
- az aktuális felhasználó hatályos engedélyével szemben az új felhasználó maximális engedélyének dominálnia kell, viszont az új felhasználó minimális engedélyével szemben dominálnia kell.

Felhasználók biztonsági felelőssége

Léteznek bizonyos felelősségek, amelyeket a felhasználóknak ismerniük, megérteniük és követniük kell. A felhasználóknak a jelszavakat titkosan kell tartaniuk, jelenteniük kell a felhasználói állapotuk változását és a gyanított biztonságsértéseket, és így tovább.

Jelszavak

A jelszavakat meg kell jegyezni, és nem szabad sehova leírni. Ha a jelszót egy másik felhasználó megszerzi, akkor ez veszélyeztetheti a rendszeren lévő információk biztonságát.

A legtöbb nyilvánvaló jelszóbiztonsággal kapcsolatos fenyegetettség a jelszavak veszélyeztetése. A legegyszerűbb módszer a fiók jogosulatlan támadás elleni védelmére olyan felhasználóval szemben, aki felderített egy jelszót, a jelszó rendszeres időközönkénti cseréje. A jelszavakat gyakran kell változtatni a veszélyeztetés valószínűségének csökkentése érdekében az egyéni jelszó élettartama során. Minél tovább használnak egy jelszót, annál nagyobb a lehetőség a veszélyeztetésre.

Ha a felhasználók kiválaszthatják a saját jelszavukat, akkor az új jelszónak legalább hat karakteresnek kell lennie és tartalmaznia kell legalább két alfanumerikus és egy numerikus karaktert. A jelszó nem tükrözheti a felhasználó személyes vagy szakmai aspektusát (például barátok, felhasználó neve, háziállat neve vagy beosztás) és nem lehet a szótárban megtalálható, általánosan használt szó. A jelszókitalálási sémák gyakran végignéznek néhány szótárat és a személyes elemek fontos listáját, mint például a felhasználó neve, a gyerek vagy háziállat neve és a születésnap.

A jelszavak véges élettartammal rendelkezhetnek, amelyet az ISSO határoz meg. Ha egy jelszó lejárt és a felhasználó megpróbál bejelentkezni, akkor a felhasználó értesítést kap, amely szerint a jelszót módosítani kell és a felhasználó bejelentkezhet, hacsak a jelszó meg nem változott. A jelszavakat a megadott jelszóélettartamnál gyakrabban érdemes módosítani. Ha gyanú merül fel a felhasználó jelszavának veszélyeztetésével kapcsolatban, akkor a jelszót azonnal meg kell változtatni.

A rendszer felügyelet nélkül hagyása

Sosem szabad a rendszert felügyelet nélkül hagyni, miközben egy felhasználó aktív munkamenetbe van bejelentkezve. Ha a géptől akár csak rövid időre is távol kell lennie, akkor ajánlott kilépni a rendszerből annak elhagyása előtt.

Biztonságos rendszerfelügyelet

A biztonságos számítógéprendszer felügyelete magában foglalja a biztonsági házirendek foganatosítása mellett a szokásos rendszermegfigyelést is.

A telephely biztonságos szolgáltatáskezelésének kifejlesztése során az alábbi lista tekinthető kiindulási pontként:

- A rendszer akkreditációs tartományában megadott maximális biztonsági szint ne legyen nagyobb a rendszert tartalmazó telephely maximális biztonsági szintjénél.

- A rendszer hardvereszközeit ajánlott védett helyen üzemeltetni. A legbiztonságosabb helyek rendszerint az olyan belső szobák, amelyek nem a földszinten helyezkednek el.
- A rendszer hardvereszközeihez való fizikai hozzáférést célszerű korlátozni, megfigyelni, illetve dokumentálni.
- A rendszer mentéseit, illetve archiválási adathordozóit ajánlott biztonságos helyen, a rendszerhardver telephelyétől távol tárolni. Ehhez a helyszínhez a fizikai hozzáférést a rendszer hardvereszközeihez való hozzáféréssel egyező módon ajánlott korlátozni.
- A működési kézikönyvekhez és adminisztrációs dokumentációhoz való hozzáférést szintén ajánlott az érvényes korlátozott hozzáférés alapján korlátozni.
- A rendszer újraindításait, áramkimaradásait, illetve leállításait ajánlott rögzíteni. A fájlrendszer sérüléseit dokumentálni kell, illetve az érintett fájlokat ajánlott elemezni, hogy tartalmaznak-e biztonsági házirend sértéseket.
- Az új programok telepítését - függetlenül attól, hogy importálás vagy létrehozás során jönnek-e létre - ajánlott korlátozni és megfigyelni. Az új programokat futtatás előtt ajánlott aprólékosan elemezni és tesztelni.
- A rendszerszoftverek szokatlan vagy nem várt viselkedését dokumentálni és jelenteni kell, illetve a viselkedés okát meg kell határozni.
- Amikor csak lehetséges, legkevesebb két ember végezze a rendszer felügyeletét. Az egyik személy rendelkezzen az isso, a másik az sa szereppel.
- A PV_ROOT jogosultság használatát ajánlott kerülni. A rendszer felügyeletéhez az ISSO, SA vagy SO felhasználók által futtatott privilegizált programoknak elegendőnek kell lenniük.
- A megfigyelési információkat tanácsos naplókban összegyűjteni, illetve rendszeresen áttekinteni. A szokatlan vagy a normálistól eltérő eseményeket tanácsos feljegyezni, illetve az események kiváltó okát felderíteni.
- Az isso, sa és so szerepekkel való bejelentkezések számát ajánlott minimálisra csökkenteni.
- A setuid és setgid programok számát ajánlott minimálisra csökkenteni, illetve a programok használatát tanácsos a védett alrendszerekre korlátozni.
- Az új programokhoz hozzárendelt jogosultságok meghatározásához és minimálisra csökkentéséhez ajánlott a meglévő programokhoz korábban hozzárendelt jogosultságokat áttekinteni.
- A fájlok és könyvtárak biztonsági attribútumait tanácsos rendszeres időközönként a **trustchk** parancs segítségével ellenőrizni.
- Minden jelszónak ajánlott legkevesebb 8 karaktert tartalmaznia. Ezt az ISSO felhasználónak tanácsos rendszeresen ellenőriznie.
- Minden felhasználóhoz tanácsos érvényes alapértelmezett bejelentkezési parancsértelmezőt rendelni. Ezt az SA felhasználónak tanácsos rendszeresen ellenőriznie.
- A normális felhasználók felhasználói azonosítói ne legyenek rendszerazonosítók. Ezt az SA felhasználónak tanácsos rendszeresen ellenőriznie. A rendszerazonosítók a 128-as uid azonosítónál kisebbel rendelkező azonosítók.

Rendszerkonfiguráció:

Bizonyos lépéseket az ISSO és SA felhasználónak végre kell hajtania a rendszer megfelelő beállítása érdekében. Az ISSO elsősorban a biztonság kezeléséért felelős, az SA pedig a napi adminisztrációért.

Az ISSO a következő feladatok hajtja végre:

- Telepíti és beállítja az alapvető biztonság funkciót, a rendszermegfigyelést, elszámolást és a lefoglalható eszközök biztonságát is beleértve.
- Szerkeszti az /etc/rc.mls és /etc/rc.mls.boot fájlban lévő rendszerindítási parancsfájlt a hely biztonsági irányelveinek való megfelelés érdekében.

  Megjegyzés: A rendszerindítási parancsfájl módosításai nem képezik a kiértékelt konfiguráció részét, ezért a rendszer akkreditálása előtt foglalkozni kell ezekkel a módosításokkal.
- Beállítja a rendszerszintű bejelentkezési paramétereket.
- Beállítja a rendszerszintű jelszóparamétereket.
- Beállítja az SL tartományt a tty eszközökhöz, amelyek lehetővé teszik, hogy a felhasználók bejelentkezzenek a tty porthoz megadott SL tartományokba. További információkért tekintse meg a **chsec** parancsot.

- Beállítja a rendszereszköz SL-eket a szalagmeghajtóhoz és hajlékonylemez-meghajtóhoz. További információkért tekintse meg a **setsecattr** parancsot.
- Beállítja a rendszer hely által beállítható szolgáltatásait.

  Megjegyzés: A beállítható biztonsági szolgáltatások módosításai nem képezik a kiértékelt konfiguráció részét, ezért a rendszer akkreditálása előtt foglalkozni kell ezekkel a módosításokkal. Az alapértelmezett konfigurációmódosítások a rendszerműködés kevésbé biztonságos módját eredményezik.
- Beállítja a megbízható biztonsági adatbázist a megbízható rendszerbetöltéshez és helyreállításhoz. További információkért tekintse meg a **trustchk** parancsot.
- Beállítja a rendszeren a felhasználói csoportokat.

Az ISSO és SA együttműködik a nyomtatók beállítása érdekében. Az SA beállítja a nyomtatót a rendszerhez, az ISSO pedig beállítja az SL tartományt a nyomtatókhoz.

Hálózatkonfiguráció:

Az ISSO elsősorban a hálózat biztonságáért, az SA pedig a hálózati adminisztrációval kapcsolatos napi teendőkért felelős. Az ISSO és az SA a hálózat megfelelő konfigurálását együttesen végzi.

A hálózati biztonság a Trusted AIX telepítése során alapértelmezett beállításokkal kerül konfigurálásra. A hálózati biztonság - ezen kívül - érzékenységi címkéket is átadhat a hálózat egyéb Trusted AIX hosztjainak. A rendszer részeként biztosított alapszintű hálózati funkcionalitás telepítését és konfigurálását az ISSO végzi. Az ISSO végzi a hálózati táblák konfigurálását, majd - a **tninit** parancs segítségével - az adatbázisok mentését.

Hálózati hozzáférés:

Ha egy hálózaton keresztül nem Trusted AIX rendszerhez vagy a Trusted Networking szolgáltatást nem használó Trusted AIX rendszerhez csatlakozik, akkor lehetséges, hogy a nem Trusted AIX rendszer bizonyos biztonsági attribútumokat nem továbbít. Ebben az esetben a Trusted AIX rendszer alapértelmezett biztonsági mechanizmusokat alkalmaz. Az alapértelmezett biztonsági mechanizmusokat a rendszeradminisztrátor hozza létre.

Felhasználói fiók beállítása:

Az ISSO és SA együtt beállítja a felhasználói fiókot a rendszeren. Az ISSO elsősorban a biztonsággal kapcsolatos felhasználói attribútumok kezeléséért, az SA pedig más felhasználói attribútumokért felelős.

Az ISSO a következő feladatokat hajtja végre minden felhasználóhoz:
- Engedély beállítása. További információkért tekintse meg a **chsec** és **chuser** parancsot.
- Szerepek és felhatalmazások beállítása
- Felhasználói csoportok beállítása
- Saját könyvtár engedélyszint beállítása. További információkért tekintse meg a **settxattr** parancsot
- Jelszó beállítása
- Megfigyelési maszkok beállítása

Az SA a következő feladatok hajtja végre:
- Felhasználói fiókok beállítása
- A biztonsági attribútumokat igénylő új felhasználói fiókok ISSO-jának informálása

Fájlrendszerek beállítása:

A Trusted AIX a legtöbb fájlrendszert támogatja, azonban a fájlrendszer-objektumokra vonatkozó Trusted AIX biztonsággal kapcsolatos kiterjesztett attribútumok csak az EAv2 fájlrendszert használó JFS2 esetén állnak rendelkezésre.

Ha az EAv1 fájlrendszert használó JFS2 fájlrendszert Trusted AIX rendszeren felépíti, akkor az átalakításra kerül EAv2 fájlrendszerré. Az ilyen JFS2 fájlrendszereken található fájlok nem rendelkeznek biztonsági attribútumokkal. A fájlok elérése során a rendszer az alapértelmezett SYSTEM_LOW attribútumokat használja. A fájlok biztonsági attribútumai a **settxattr** parancs segítségével állíthatók be.

Hálózati környezetekben az egyik rendszeren található könyvtár megjelölhető megosztottként, vagyis a könyvtár úgy építhető fel és érhető el a hálózat egyéb rendszerein, mintha egy helyi lemezpartíció gyökérkönyvtára lenne.

A fájlrendszerek lehetnek többszintű (MLFS), illetve egyszintű (SLFS) fájlrendszerek. Az MLFS fájlrendszereken minden fájlobjektum saját címkével rendelkezik, ezzel szemben az SLFS objektumai mind a felépítési ponttal egyező címkével rendelkeznek. Az SLFS a többszintű és particionált könyvtárakat nem támogatja.

Fájlrendszer-hozzáférés:

Amikor egy folyamat kísérletet tesz egy fájlrendszer-objektum elérésére, akkor a rendszer ellenőrzi az egyes útvonalnév-összetevőkhöz való hozzáférést.

Ha a folyamat nem rendelkezik keresési hozzáféréssel az útvonalnévben megadott összes könyvtárhoz, akkor az objektumhoz a folyamat nem férhet hozzá. Relatív útvonalnév használatakor az aktuális könyvtárhoz való hozzáférés kerül ellenőrzésre, attól függetlenül, hogy az útvonalnév elején hivatkozik-e kifejezetten az aktuális könyvtárra a pont (.) használatával.

Trusted Network kezelése:

A Trusted Network kezeléséhez számos tényező áll rendelkezésre, a konfigurációt, a konfigurációs adatbázist, netrule szintaxist és szabályspecifikációt, a Trusted Network jelzőket és a RIPSO/CIPSO paramétereket is beleértve.

Alapértelmezett konfiguráció figyelmeztetés:

Az AIX Trusted Network hálózatkezelési képességét úgy tervezték, hogy lehetővé tegyen bármilyen elképzelhető konfigurációt. Azonban az alapértelmezett értékek módosítása - az AIX Trusted Network hálózatok alapos ismeretének hiányában - veszélyes lehet.

A számítógép nem megfelelő beállításának következtében előfordulhat, hogy a biztonsági információk kiemelésre, visszaléptetésre vagy teljesen eltávolításra kerülnek. Ennek következtében nem ajánlott a hálózatkezelési táblákban található alapértelmezett értékek módosítása, hacsak nem ismeri alaposan az AIX Trusted Network hálózatot.

AIX Trusted Network konfigurációs adatbázis:

A rendszerbetöltés során a hálózati konfigurációt a rules.host és rules.int fájlok hozzák létre.

Egy alapértelmezett Trusted AIX telepítés után nincsenek hoszt szabályok és szabályfájlok. Ha az új vagy frissített szabályokat fájlba kívánja menteni, akkor használja a **netrule** parancsot az **-u** kapcsolóval. A fájlok bináris adatbázisok, amelyek a **tninit** parancs segítségével kezelhetők. A **tninit** parancs használatához a felhasználónak aix.mls.network.init felhatalmazással kell rendelkeznie.

AIX Trusted Network szabályadatbázis megjelenítése:

Az AIX Trusted Network szabályadatbázisa a **tninit** parancs **disp** tevékenységével jeleníthető meg.

Ha a **.host** és **.int** kiterjesztéseket hozzá kívánja fűzni a *fájlnév*hez és ezáltal előállítani a hosztszabály- és csatolószabály-adatbázis fájlnevét, akkor adja ki a következő parancsot. A két fájl tartalma a felhasználó által olvasható formátumban elküldésre kerül a szabványos adatfolyam kimenetre.

```
tninit disp fájlnév
```

A rendszerbetöltési alapértelmezett konfiguráció megjelenítéséhez adja ki a következő parancsot:

```
tninit disp /etc/security/rules
```

AIX Trusted Network szabályadatbázis betöltése:

A **tninit** parancs beolvas egy AIX Trusted Network szabályadatbázis-halmazt, majd betölti azt a kernelbe, hogy ezáltal ez legyen az aktív halmaz. A hoszt- és csatolóakkreditálási táblák a **tninit disp** tevékenységével egyező módon kerülnek meghatározásra.

Az elhagyható **-m** kapcsoló meghatározza, hogy a rendszer a meglévő hosztszabályokat tartsa meg. Ha az **-m** kapcsolót nem adja meg, akkor az új aktív halmaz betöltése előtt az összes meglévő hosztszabály eltávolításra kerül. Ha az **-m** kapcsolót megadja, akkor a meglévő és új hosztszabályhalmazok összesítésre kerülnek; ütközés esetén az új szabályok a meglévő szabályokat felülírják. Attól függetlenül, hogy az **-m** kapcsolót megadja-e, a csatolószabályok mindenképpen felülírásra kerülnek.

A következő parancs a régi szabályhalmaz megtartása mellett betölti az új szabályokat:

```
tninit -m load /dir/dir/fájlnév
```

A parancs a *fájlnév* paraméterrel megadott fájlnevet használja, majd hozzáfűzi a fájlnévhez a **.host** és **.int** kiterjesztéseket, hogy ezáltal létrehozza az adatbázist alkotó két fájlt.

AIX Trusted Network szabályadatbázis mentése:

A szabályadatbázis betöltése és mentése során hasonló szemantika használható.

A tetszőlegesen megadott fájlnév kiegészítésre kerül az **.int** és **.host** utótaggal, hogy ezáltal létrehozza az adatbázis tárolásához használt két fájlt. A **tninit** parancs mentési tevékenysége az összes kernelben pillanatnyilag aktív szabályt eltárolja.

Az alapértelmezett szabályhalmaz létrehozásához a **netrule** parancs segítségével a telephely biztonsági házirendjének megfelelően módosítsa a kernelszabályokat, majd futtassa a **tninit** parancsot. A következő parancs létrehozza az /etc/security/rules.int és az /etc/security/rules.host fájlokat:

```
tninit save
/etc/security/rules
```

AIX Trusted Network kernel beállítása:

Ha aix.mls.network.config jogosultsággal rendelkezik, akkor a **netrule** parancs segítségével a telephely biztonsági házirendjének megfelelően teljes körűen beállíthatja a kernel AIX Trusted Network szabályhalmazát.

A kernelben a hoszt és a hálózati csatoló szabályai a **netrule** parancs segítségével kezelhetők. További információkat a **netrule** parancs leírása tartalmaz.

A rendszer minden csatolójához szabályhalmaznak kell tartoznia. Ha egy csatolószabályt megpróbál törölni, akkor a csatoló visszaáll alapértelmezett állapotába. Ha új csatolószabályt vesz fel, akkor az új szabály felülírja az aktuális szabályt. Az alapértelmezett csatolószabály megjelenítéséhez a csatolószabályt a “default” csatolónév használatával kérdezze le. Például: # netrule iq default

netrule szintaxis:

A **netrule** parancs külön hoszt és csatoló szintaktikai szabályokkal rendelkezik.

Hosztok esetében a **netrule** parancs az alábbi szintaktikai szabályokkal rendelkezik:

netrule h l [**i** | **o** | **io**]

netrule h q { **i** | **o** } *forrás_hosztszabály-meghatározás cél_hosztszabály-meghatározás*

netrule h - [{ **i** | **o** } [**u**] [*forrás_hosztszabály-meghatározás cél_hosztszabály-meghatározás*]

netrule h + { **i** | **o** } [**u**] *forrás_hosztszabály-meghatározás cél_hosztszabály-meghatározás* [*kapcsolók*] [*RIPSO/CIPSO_beállítások*] *biztonság*

Csatolók esetében a **netrule** parancs az alábbi szintaktikai szabályokkal rendelkezik:

netrule i l

netrule i q *csatoló*

netrule i + [**u**] *csatoló* [*kapcsolók*] [*RIPSO/CIPSO_beállítások*] *biztonság*

Az első elem - h vagy i - a hoszt vagy hálózati csatoló műveletet jelöli.

A listában a következő a kívánt tevékenység. Négy különböző tevékenység áll rendelkezésre:

l Összes szabály listázása

q Adott szabály lekérdezése

- Hosztszabály eltávolítása vagy csatolószabály visszaállítása az alapértelmezett állapotra

+ Szabály felvétele vagy felülírása

A hosztszabályok harmadik eleme a szabálytípust azonosítja. A hosztszabályok esetében különbséget kell tenni a bejövő és kimenő szabályok között. A bejövő szabályok az összes bejövő csomagra vonatkoznak, a kimenő szabályok pedig az összes kimenő csomagra. A bejövő szabályokat i, a kimenő szabályokat o jelöli, illetve - ha alkalmazható - io vagy semmi jelöli a ki- és bemenő szabályokat. Ha az utolsó u elemet hoszt- vagy csatolószabály felvételekor vagy eltávolításakor megadja, akkor a hoszt- vagy csatolószabály sikeres felvételét vagy eltávolítását követően az /etc/security/rules.host és /etc/security/rules.int fájlok frissítésre kerülnek.

AIX Trusted Network szabályok meghatározása:

A csatolószabályok megkövetelik a hálózati csatoló nevének megadását. A hosztszabályok ezeknél sokkal rugalmasabbak, és ezért összetettebb szabálymeghatározást igényelnek.

A csatoló meghatározásához adja meg annak a hálózati csatolónak a nevét, amelyre a szabály vonatkozik. A hálózati csatoló neve lehet például en0 vagy ehhez hasonló. A hálózati csatolók neve az **ifconfig -a** parancs segítségével jeleníthető meg. A csatoló nevével adjon meg egy adott csatolót. Port, protokoll és alhálózati maszk nem adható meg.

A hosztszabályok összetettebb szabálymeghatározást igényelnek. Az AIX Trusted Network rendszer a legpontosabban meghatározott, alkalmazható szabályt használja. Egy telephely házirendje beállítható például úgy, hogy a 24-es maszkkal rendelkező hosztszabály az alhálózat összes hosztjára vonatkozik, de egy pontosabban meghatározott szabály csak a hálózat egyetlen hosztjára vonatkozik. Ez a hoszt a pontosabban meghatározott szabályt használja. Egy másik pontosabban meghatározott szabály vonatkozhat a hoszt egy adott TCP portjára. Az AIX Trusted Network beállítása nyújtotta rugalmasság lehetővé teszi az alkalmazáshoz szükséges tetszőleges biztonsági házirend megvalósítását. A pontos szintaxis:

forrás_hoszt [/*maszk*] [**=** *proto*] [**:***porttartomány_kezdete* [**:***porttartomány_vége*]]

cél_hoszt [/*maszk*] [**=** *proto*] [**:***porttartomány_kezdete* [**:***porttartomány_vége*]]

forrás_hoszt
A forrás hoszt hosztneve, IPv4 címe vagy IPv6 címe.

cél_hoszt
A cél hoszt hosztneve, IPv4 címe vagy IPv6 címe.

maszk Az alhálózati maszk. A szám jelöli, hogy az MSB hány bitje fontos. Ha egy IPv4 cím/alhálózat pár *a***.***b***.***c***.***d*/*e*

formátumú, akkor az *e* egy 0-32 közötti szám. A szám megadja az alhálózati maszk elején található egyesek számát. IPv4 cím esetében például a /24 a 255.255.255.0 hálózati maszkot adja meg, amely - 32 bites átírásban - 11111111.11111111.11111111.00000000. Erre utal a 24 egyes, majd nyolc nulla.

proto
Az /etc/protocols fájlban rögzített protokollszám és -név (például: =tcp).

porttartomány_kezdete
A TCP vagy UDP port, amelyre a szabály vonatkozik, vagy - amennyiben a szabály porttartományra vonatkozik - a porttartomány kezdete. Az érték lehet a port száma vagy az /etc/services fájlban rögzített UDP vagy TCP szolgáltatás neve.

porttartomány_vége
A porttartomány felső határa.

AIX Trusted Network kapcsoló leírása:

Az AIX Trusted Network rendszer két kapcsolócsoporttal rendelkezik. Ha ezeket nem adja meg, akkor az alapértelmezett értékek kerülnek felhasználásra.

A **-d** és **-r** kapcsolók használata:

-d *eldobás*

eldobás
Az AIX Trusted Network beállítható úgy, hogy az összes csomagot eldobja

r A csatoló összes csomagjának eldobása

n Ne dobja el automatikusan a csatoló összes csomagját (csatoló alapértelmezése)

i A csatoló alapértelmezésének használata (hoszt alapértelmezése, csak hoszt esetén)

-f*rkapcsoló***:***tkapcsoló*

rkapcsoló
A bejövő (fogadott) csomagokra vonatkozó biztonsági beállítás követelmény

r Csak RIPSO

c Csak CIPSO

e CIPSO vagy RIPSO

n Sem CIPSO, sem RIPSO (rendszer alapértelmezése)

a Nincs megszorítás

i Csatoló/rendszer alapértelmezésének használata (alapértelmezés)

tkapcsoló
A kimenő (átvitt) csomagok kezelését szabályozó biztonsági beállítás

r A kimenő csomagok IP fejléce RIPSO beállítással egészül ki

c A kimenő csomagok IP fejléce CIPSO beállítással egészül ki

i A csatoló alapértelmezésének használata (hoszt alapértelmezése, csak hoszt esetén)

RIPSO/CIPSO beállítások:

A AIX Trusted Network alrendszer támogatja a CIPSO és RIPSO csomagcímkézés beállítására szolgáló beállításokat.

-rpafs=*PAF_mező* [**,** *PAF_mező* ...]

Meghatározza az IPSO csomagok fogadásakor elfogadott *PAF_mező*ket. Legfeljebb 256 ilyen mező adható meg.

-epaf=*PAF_mező*

Meghatározza azt a *PAF_mező*t, amely a hibaválaszokhoz csatolásra kerül akkor, ha a hibacsomagok az átvitt csomagokon IPSO felhasználásával kerülnek küldésre.

-tpaf=*PAF_mező*

Meghatározza a kimenő csomagokra vonatkozóan alkalmazni kívánt *PAF_mező*t, amikor az átvitt csomagokon IPSO kerül felhasználásra.

*PAF_mező***:NONE |** *PAF* [**+** *PAF ...*]

A *PAF_mező* több *PAF* gyűjteménye. Öt olyan egyedi *PAF* létezik, amelyeket egy *PAF_mező* tartalmazhat. Ezek a következők: **GENSER**, **SIOP-ESI**, **SCI**, **NSA** és **DOE**. A *PAF_mező*k ezen értékek egyesítései, egy plusz jel (**+**) elválasztásával. A **GENSER** és **SCI** PAF-okat egyaránt tartalmazó *PAF_mező*t **GENSER+SCI** jelöli. A különleges **NONE** *PAF_mező* is használható; ez a beállított *PAF* nélküli *PAF_mező*t jelöl.

-DOI=*doi*

Meghatározza a CIPSO csomagok értelmezési tartományát (DOI). A bejövő CIPSO csomagoknak rendelkezniük kell ezzel a **DOI** értékkel, illetve a kimenő csomagok megkapják ezt a **DOI** címkét.

-tags=*jelölő*[**,***jelölő ...*]

*jelölő***=1 | 2 | 5**

Meghatározza a CIPSO beállítások által elfogadott, illetve az átvitelhez rendelkezésre álló jelölők halmazát. Ez az **1**, **2** és **5** értékek vesszővel elválasztott sora. Az **1,2** például az **1** és **2** jelölők használatát engedélyezi.

AIX Trusted Network biztonsági irányelv:

Meg kell határozni a minimálisan megengedett, a maximálisan megengedett, illetve az alapértelmezett SL címkét.

Az implicit vagy alapértelmezett SL az összes olyan csomagra vonatkozik, amely saját SL címkéjére vonatkozóan információkat nem tartalmaz. A szintek az alábbi szintaxis segítségével adhatók meg:

+*min* **+***max* **+***alapértelmezett*

A címkekódolási fájl szerint érvényes valamennyi címke használható. A szóköz karaktereket tartalmazó címkék esetében az idézőjel használata nem kötelező.

netrule példák:

Az alábbi példák a **netrule** használatát mutatják be.

Ha az **en0** hálózati csatolót úgy kívánja beállítani, hogy biztonsági beállításokat ne adjon át, illetve az összes csomagot átengedje, akkor adja ki a következő parancsot:

```
netrule i+ en0 +impl_lo +ts all +impl_lo
```

Ha a **185.0.0.62** hosztot úgy kívánja beállítani, hogy csak a **CONFIDENTIAL A** és **TOP SECRET ALL** tartományba eső CIPSO csomagokat fogadja el, akkor adja ki a következő parancsot:

```
netrule h+i
192.168.0.0 /24 185.0.0.62 -fc:c +confidential a +top secret all +confidential a
```

Ha az egyik alhálózatból érkező összes telnet csomagot el kívánja dobni, akkor adja ki a következő parancsot:

```
netrule h+i 192.168.0.0 /24
=tcp :telnet 192.0.0.5 -dr +impl_lo +impl_lo +impl_lo
```

További információkat és példákat a **netrule** parancs leírása tartalmaz.

Felhasználói fiókok kezelése:

Az egyes felhasználókkal kapcsolatos azonosítási és hitelesítési (I&A) információi védettek és egyedien azonosítják a felhasználókat, illetve ellenőrzik a felhasználó rendszeren belüli hozzáférési jogait.

A felhasználóazonossági információk a következőket tartalmazzák: a felhasználó neve, bejelentkezési azonosító szöveges neve, felhasználói azonosító, csoportazonosító, saját könyvtár, jelszó, jelszóöregedési paraméterek, parancsértelmező, engedélyek, felhatalmazások és megfigyelési maszk. A legtöbb felhasználóval kapcsolatos információt a következőt fájlok tárolják:

/etc/passwd
: Felhasználónevek, felhasználói azonosítók, elsődleges csoport-hozzárendelések és saját könyvtárak

/etc/group
: Másodlagos csoporthozzárendelések és saját könyvtárak

/etc/security/passwd
: Felhasználói jelszavak titkosított formában

/etc/security/user
: Bejelentkezési korlátozások, jelszóparaméterek (mint például a minimális hossz), umask, stb.

Az /etc/security/passwd és /etc/security/user fájlt normál felhasználók nem olvashatják. Az /etc/security/passwd fájlt bekapcsolt tetszés szerinti hozzáférési bitek és a SYSTEM_HIGH SL-je védi. Annak megakadályozása, hogy a normál felhasználók olvashassák a titkosított jelszót, megszünteti a sorozatos titkosítási/összehasonlítási rutinokat, amelyek a titkosított jelszó egyeztetését kísérlik meg.

A felhatalmazott felhasználók közvetlenül szerkeszthetik ezeket a fájlokat, de gyakran az **smit** paranccsal kényelmesebben szerkeszthetők a felhasználói paraméterek. Az **smit** parancs meghívja az SMIT-t, amely menüket jelenít meg lehetőségekkel a rendszerfelügyeleti feladatokhoz, mint például a felhasználó-karbantartás.

Felhasználói és csoportazonosító:

Két felhasználói azonosító osztály létezik: rendszerazonosítók és normál felhasználói azonosítók. A rendszerazonosítók védett alrendszerek birtoklásához és speciális rendszeradminisztrátori funkciók számára vannak fenntartva. A normál felhasználói azonosítók a rendszert interaktív módon használó egyénekhez vannak rendelve.

Minden felhasználó egyedi felhasználói azonosítóval rendelkezik a rendszer felhasználójának azonosításához. Minden felhasználóhoz egy vagy több csoportosítú rendelhető. A csoportazonosítókat a csoportban lévő felhasználók megosztják és nem feltétlenül kell egyedinek lenniük. Az azonosítókhoz használt számértékekre tartománykorlát vonatkozik. A következő táblázat az azonosító tartománykorlátját adja meg. Elegendő számú rendszer- és normál felhasználói azonosítót, valamint csoportazonosítót lehetővé tevő értékek kerültek megadásra.

Rendszerfelhasználói azonosító
: 0 - 127

Normál felhasználói azonosító
: 128 - MAXUID

Normál csoportazonosító
: 0 - MAXUID-1

A MAXUID érték az /usr/include/sys/param.h fájlban van megadva

Körültekintően kell felhasználói azonosító értékeket új felhasználókhoz rendelni. Ha egy normál felhasználóhoz véletlenül 128-nál kisebb felhasználói azonosító lett rendelve, akkor a felhasználó nem fog tudni bejelentkezni a rendszerre.

A felhasználói azonosító értékek nem használhatók fel újra. A felhasználó törlésekor ajánlatos az /etc/passwd és /etc/security/passwd fájlban maradó bejegyzéseket és a fiókot zárolni. Ez az **smit** paranccsal hajtható végre. Ez megakadályozza a felhasználó bejelentkezését és az azonosító újrafelhasználását. Az, hogy az azonosító nem kerül újrafelhasználásra, megakadályozza, hogy egy új felhasználó elérje a korábbi felhasználóhoz tartozó fájlokat, amelyek nem lettek eltávolítva. Ez lehetővé teszi a nyomkövetési napló újbóli összeállítását kétértelműség nélkül.

Az /etc/passwd, /etc/security/passwd és /etc/group fájl az **mkuser**, **chuser**, **rmuser**, **pwdadm** és **passwd** paranccsal kezelhető. Ezek a parancsok kikényszerítik a fenti óvintézkedéseket, valamint más rendszerbiztonsági megfontolásokat. Az **mkuser** parancs csak normál felhasználókat tud a rendszerhez adni.

Megjegyzés: Körültekintően tartassa be a következőket:

- Sose rendelje hozzá a korábbi felhasználó felhasználói azonosítóját egy új felhasználóhoz
- Sose rendeljen hozzá többszörös felhasználói azonosítókat
- Sose rendeljen rendszerazonosítót normál felhasználóhoz
- Sose rendelje hozzá a MAXUID-t felhasználói vagy csoportazonosítóként

Jelszavak:

A jelszó egy felhasználóhoz tartozó karaktersorozat, amelynek segítségével a felhasználót a munkamenet kezdetekor a rendszer hitelesíti.

A jelszót a rendszer titkosított formában tárolja az árnyék fájlban. A nem titkosított jelszó a rendszeren nem kerül tárolásra.

Megjegyzés: A szereppel rendelkező felhasználók jelszava a rendszer biztonsága szempontjából kulcsfontosságú, tehát azt folyamatosan védeni kell.

Jelszó öregedése:

A felhasználók a jelszó-öregedési feltételek figyelembevételével jelszavukat szabadon módosíthatják.

A jelszóöregedés megköveteli, hogy a felhasználók jelszavukat megváltoztassák akkor, ha a jelszó a rendszeren egy meghatározott időtartamon keresztül létezett. A jelszóöregedés magában foglalja a minimális és maximális öregedési időtartamot. A jelszavak a minimális öregedési időtartamon belül nem módosíthatók. A jelszavakat a maximális öregedési időtartam letelte után kötelező módosítani.

A jelszó-öregedési paraméter az /etc/security/user fájlban állíthatók be. A jelszóöregedéssel kapcsolatos paraméterek:

maxage
A jelszó érvényességének maximális hossza (hetekben)

maxexpired
A maxage utáni hetek maximális száma, amikor a felhasználó jelszavát még módosíthatja

minage
A jelszómódosítások között eltelt hetek minimális száma

minlen A jelszavak minimális hossza

A jelszavakban használható karaktereket meghatározó egyéb beállítható paraméterek. A jelszóparaméterek teljes listáját a **passwd** parancs leírása tartalmazza.

Parancsértelmező:

Miközben a felhasználók egy alkalmazásban - például szövegszerkesztőben vagy táblázatkezelőben - dolgoznak, rendszerint nincs szükségük arra, hogy az operációs rendszerrel közvetlenül dolgozzanak, mivel ezt az interakciót az alkalmazás kezeli. Azonban bizonyos felhasználók számára szükséges, hogy közvetlenül, egy másik alkalmazás felülete nélkül tudjanak az operációs rendszerrel dolgozni.

Amikor az operációs rendszerrel közvetlen interakció szükséges, akkor a felhasználóknak parancsértelmező programot kell használniuk. A parancsértelmező programok segítségével a felhasználók AIX parancsokat adhatnak ki, fájlokat és könyvtárakat közvetlenül érhetnek el, illetve egyéb műveleteket végezhetnek. Minden felhasználó számára meg kell adni egy parancsértelmező programot a felhasználóhoz tartozó /etc/passwd fájlban. A felhasználó alapértelmezett parancsértelmező programját (például /bin/sh, /bin/csh vagy /bin/ksh) a **login** vagy **xterm** parancs futtatja akkor, amikor a felhasználónak parancsértelmező használatára van szüksége.

Bejelentkezési tényleges SL és TL:

A felhasználókhoz hozzárendelésre kerül egy alapértelmezett bejelentkezési SL és TL. Az alapértelmezett bejelentkezési SL és TL a felhasználói folyamat tényleges SL és tényleges TL címkéje a sikeres bejelentkezést követően.

Ha a felhasználó nem az alapértelmezett bejelentkezési SL használatával kíván bejelentkezni, akkor bejelentkezéskor a **login** parancs **-e** paraméterével eltérő SL címkét jelölhet ki. A felhasználó által megadott SL címkével szemben a felhasználó engedélyének dominálnia kell, illetve a címkét tartalmaznia kell a rendszer akkreditációs tartományának. A TL bejelentkezéskor a **login** parancs **-t** paraméterével adható meg.

Az alapértelmezett bejelentkezési SL és TL meghatározását az /etc/security/user fájl tartalmazza, az egyes felhasználók felhasználónevével és engedélyével. A felhasználó tényleges SL címkéjének az /etc/security/login.cfg fájlban megadott tty SL tartományba kell esnie. A tty maximális SL címkéjének a felhasználó tényleges SL címkéjével szemben dominálnia kell, illetve a tényleges SL címkének dominálnia kell a minimális SL címkével szemben. A felhasználó tényleges TL címkéjének meg kell egyeznie a tty TL címkéjével.

Engedélyek:

A bejelentkezés során a felhasználó folyamat-parancsértelmezőjéhez hat címke kerül hozzárendelésre.

A tényleges SL címkét a rendszer a MAC ellenőrzések során használja. A minimális és maximális SL engedély korlátozza a tényleges SL címkét; a tényleges SL nem dominál a maximális SL engedéllyel szemben, de meg kell határoznia a minimális SL engedélyt. A tényleges TL címkét a rendszer a MIC ellenőrzések során használja. A minimális és maximális TL engedély korlátozzák a tényleges TL címkét; a tényleges TL nem határozhatja meg a maximális TL engedélyt, de meg kell határoznia a minimális TL engedélyt.

Az ISSO felhatalmazással rendelkező felhasználók tetszőleges felhasználó SL és TL engedélyét, illetve alapértelmezett bejelentkezési SL és TL címkéjét is módosíthatja. Az értékek meghatározását az /etc/security/user fájl tartalmazza.

Felhasználói információkkal kapcsolatos felelősség felosztása:

Egyetlen felhasználó a rendszerhez felhasználót nem adhat hozzá. A felhasználók hozzáadása a rendszerhez az SA és ISSO jogosultsággal rendelkező felhasználók egyesített tevékenysége.

Az SA jogosultsággal rendelkező felhasználó csak a biztonsággal nem kapcsolatos felhasználói információkat vehet fel, például a felhasználó nevét, a felhasználói azonosítót, a csoportazonosítót, a bejelentkezési azonosító szöveges nevét, a parancsértelmezőt és a saját könyvtárat. Az ISSO jogosultsággal rendelkező felhasználók csak biztonsággal kapcsolatos felhasználói információkat vehetnek fel, például a felhasználó jelszavát, jogosultságait, megfigyelési maszkját, illetve szerepeit. Az a követelmény, hogy a felhasználó felvételéhez két ember szükséges, megakadályozza, hogy egy jogosult felhasználó egy másik felhasználónak rendszerszintű jogosultságokat adjon.

Kiterjesztett megfigyelés:

A Trusted AIX megfigyelési alrendszere kiterjesztésre került további biztonsági részletek lementésével.

Új megfigyelési rekordmezők:

A Trusted AIX esetében az AIX megfigyelési rekordjai az alábbi mezőkkel bővültek. Az új mezők az **auditselect** parancsban kijelölési feltételként használhatók.

- A megfigyelt folyamat szerepei
- A megfigyelt folyamat vagy objektum tényleges TL címkéje
- A megfigyelt folyamat vagy objektum tényleges SL címkéje
- A megfigyelt folyamat tényleges jogosultságai

A Trusted AIX bizonyos nyomkövetési naplókban - az előzőek mellett - az alábbi biztonsági attribútumokat is megfigyeli:

- A megfigyelt folyamat vagy objektum TL címkéje
- A megfigyelt folyamat vagy objektum SL címkéje
- A Trusted AIX rendszerhez kapcsolódó biztonsági kapcsolók

Az új biztonsági attribútumok az **auditpr -v** parancs segítségével jeleníthetők meg.

Megfigyelési tartományok:

A Trusted AIX tartalmaz egy olyan mechanizmust, amely lehetővé teszi az adminisztrátorok számára, hogy a megfigyelt folyamatok vagy objektumok TL és/vagy SL címkéje alapján egy sor megfigyelési tartományt határozzanak meg. A megfigyelési tartományon kívül eső TL vagy SL címkével rendelkező objektumok és alanyok figyelmen kívül maradnak.

Ha a folyamatokhoz és objektumokhoz megfigyelési tartományt kíván beállítani, akkor az /etc/security/audit/config fájlhoz adjon hozzá egy **war** szakaszt:

```
war:
        obj_min_sl = "impl_lo a,b"
        obj_max_sl = "TS a,c"
        sub_min_sl = "impl_lo a,b"
        sub_max_sl = "TS a,c"
        obj_min_tl = impl_lo
        obj_max_tl = TS
        sub_min_tl = impl_lo
        sub_max_tl = TS
```

Az objektumok SL megfigyelési tartományát az **obj_min_sl** és **obj_max_sl** határozza meg. Az alanyok (folyamatok) SL megfigyelési tartományát a **sub_min_sl** és **sub_max_sl** határozza meg. Az objektumok TL megfigyelési tartományát az **obj_min_tl** és **obj_max_tl** határozza meg. Az alanyok (folyamatok) TL megfigyelési tartományát a **sub_min_tl** és **sub_max_tl** határozza meg.

A **war** szakaszt az **audit start** parancs olvassa be, majd a megfigyelési alrendszer indítása előtt a szakasz feltöltésre kerül a kernelbe. Ha a **war** szakasz figyelmen kívül marad, akkor a kernel aktuális megfigyelési tartományai eltávolításra kerülnek. Ha a kernel TL SL megfigyelési tartományt nem tartalmaz, akkor a kernel TL és SL megfigyelésitartomány-ellenőrzéseket nem hajt végre.

Trusted AIX kernel kapcsoló:

Ha telepítéskor egy rendszert Trusted AIX rendszerként állít be, akkor a **_system_configuration** változóban egy globális kernel kapcsoló beállításra kerül. Annak meghatározására, hogy a rendszer Trusted AIX rendszer-e, a kernel **__MLS_KERNEL()** makrója szolgál. A makrót meghívhatják a felhasználói alkalmazások, illetve a kernelrutinok. Az

__MLS_KERNEL() makró **1** visszatérési értéke jelöli, hogy a rendszer Trusted AIX rendszerként került beállításra. Az ettől eltérő visszatérési értékek azt jelölik, hogy a rendszer nem került Trusted AIX rendszerként beállításra.

Meglévő programok frissítése:

A meglévő privilegizált és megbízható programok az esetek többségében a megbízható rendszereken minden módosítás nélkül megfelelően működnek.

Azonban bizonyos módosítások végrehajthatók a megbízhatóság kiterjesztése, illetve a programok későbbi változatokkal való kompatibilitásának érdekében. Számos, az új programokra vonatkozó ajánlás vonatkozik a meglévő programok frissítésére is. Az ajánlások közül különösképpen fontosak az alábbiak:

- Az olyan programokat, amelyek tesztelés útján határozzák meg, hogy privilegizált folyamatok-e (tehát azt, hogy a hatályos felhasználói azonosító 0-e), a Közvetlen jogosultságellenőrzés szakaszban leírt irányelvek alapján módosítani kell.
- A szabványos UNIX rendszerjogosultsági biteket (módbiteket) kezelő kódokat módosítani kell, hogy tükrözzék az ACL listák esetleges létezését
- Az olyan kódokat, amelyek futtatása során a setuid-to-root vezérlőbit beállított volt, meg kell vizsgálni, majd a megfelelő jogosultságokat hozzájuk kell rendelni

Mentés és visszaállítás:

A Trusted AIX rendszereken az adatok importálása és exportálása a **backup** és **restore** parancsok megbízható változatainak segítségével történik.

A **backup** és **restore** parancsok kiterjesztésre kerültek a címkék kezelése érdekében. A kiterjesztések a felhasználók számára egyértelműek, illetve - a címkekezelési kiterjesztés kivételével - a parancsok az általános AIX **backup** és **restore** parancsokkal egyező módon működnek. A kiterjesztett biztonsági információk mentésének vagy visszaállításának letiltására az **-O** kapcsoló használható.

Az importálási/exportálási rendszert a jogosultsági és felhatalmazási mechanizmusok együttese védi.

cron korlátozások:

A rendszer konfigurációs módjában a **cron** parancs tiltott és feladatokat nem futtat. A rendszer működési módjában a **cron** parancs a feladatot azon az érzékenységi címkén futtatja, amelyen a feladat benyújtásra került, illetve amely a felhasználó alapértelmezett integritás címkéje.

Léteznek korlátozások, például a felhasználó minimális és maximális engedélye. Attól függően, hogy melyik az újabb, az engedély vagy a feladat benyújtásának, vagy a **cron** parancs utolsó újraindításának időpontjában megadott beállításokból származik. A **cron** parancsot csak SA felhasználók kezelhetik.

Fájlrendszerek felépítése és elérése:

Az EAv2 fájlrendszert használó JFS2 fájlrendszer esetében a Trusted AIX támogatja a címkék (SL és TL) használatát. Ha szükséges, akkor az SA vagy SO felhasználók a címkék használatát nem támogató (CDFS vagy HSFS) fájlrendszert is felépíthetnek. Ebben az esetben a felépített fájlrendszeren található fájlok nem rendelkeznek egyéni SL és TL címkékkel, illetve FSF kapcsolókkal, hanem a felépítési pont biztonsági attribútumait öröklik.

Trusted AIX rendszerfelügyelet

A Trusted AIX rendszer megfelelő felügyeletének irányelveit - a rendszerbiztonság érdekében - ajánlott követni.

A Trusted AIX rendszerfelügyeletet az adminisztrátori szerephez tartozó fiókkal rendelkező felhasználók végzik. Az ilyen felhasználók az ún. információrendszer adatvédelmi megbízottak (ISSO), a rendszeradminisztrátor (SA), illetve a rendszerfelelős (SO). A felhasználók - jogosultságaiknak köszönhetően - az adminisztrációs feladatok egy-egy adott részhalmazát végezhetik el. A felhasználók a rendszer által meghatározott isso, sa, illetve so szerephez tartoznak, értelemszerűen. Az ISSO, SA és SO kifejezések az isso, sa, illetve so szereppel rendelkező felhasználókra utalnak,

értelemszerűen. Bizonyos adminisztrációs feladatokat csak kettő vagy több rendszerkezelő együttesen végezhet el, mivel a kezelők külön-külön szinte sosem rendelkeznek elegendő jogosultsággal a feladatok végrehajtásához. A rendszerre új felhasználó felvételekor például csak az SA vehet fel új felhasználói fiókot, illetve csak az ISSO hozhatja létre a felhasználó jelszavát, jogosultságait és megfigyelési maszkját. Ezt a munkamegosztást más néven kétfős szabálynak hívjuk.

Megjegyzés: A kétfős szabály hatékonysága az adminisztrátori szerepekhez rendelt felhatalmazásoktól függ. Az adminisztrátori szerepekhez a szükségesnél több felhatalmazás hozzárendelése a rendszert a belső támadásokkal szemben sérülékennyé teheti. A felhatalmazások szerepekhez rendelésével kapcsolatosan további információkat az RBAC témakör tartalmaz.

Alapértelmezésben a rendszer által meghatározott isso, sa, illetve so szerepekhez az alábbi Trusted AIX jogosultságok tartoznak. Különösen körültekintően kell eljárni a hozzárendelések módosítása során, mivel ez a rendszert sérülékennyé teheti.

41. táblázat: Szerepek és jogosultságok

isso	sa	so
		aix.mls.login
	aix.mls.printer	
aix.mls.network.config		
aix.mls.network.init		
aix.mls.network.config		
aix.mls.login		
aix.mls.pdir		
aix.mls.system.label		
aix.mls.tpath		
aix.mls.label		
aix.mls.system.config		
aix.mls.proc		
aix.mls.clear		
aix.mls.lef		
aix.mls.stat		
aix.mls.printer		

Rendszerfelügyelet információrendszer adatvédelmi megbízottak számára:

A Trusted AIX rendszerek felügyeletét az ISSO, SA és SO felhasználók összehangolt tevékenysége biztosítja.

A Trusted AIX telepítése során három alapértelmezett felhasználói fiók (**isso**, **sa** és **so**) kerül létrehozásra, hacsak ezek a fiókok már nem léteznek a rendszeren (például normál AIX rendszerről Trusted AIX rendszerre áttérés során). A felhasználók az isso, sa, illetve so szerephez tartoznak, értelemszerűen.

Megjegyzés: Az alapértelmezett fiókok csak a Trusted AIX rendszerek kezdeti beállítására és konfigurációjára szolgálnak. Ajánlott ezeket a szerepeket egyéb normál felhasználókhoz rendelni. Miután a szerepek más felhasználókhoz hozzárendelésre kerültek, az alapértelmezett felhasználói fiókok eltávolíthatók. A Trusted AIX telepítésével kapcsolatosan további információkat az *Installation and migration* tartalmaz.

ISSO tevékenységek

Az információrendszer adatvédelmi megbízott elsődleges feladata a rendszerbiztonság adminisztrációs feladatainak elvégzése. ISSO tevékenységeket csak az ISSO jogosultsággal rendelkező felhasználók végezhetnek. Az ilyen tevékenységek - többek között - az alábbiak:

- A telephely biztonsági házirendjének tervezése, megvalósítása, illetve betartatása
- A felhasználói engedélyek, felhatalmazások, jogosultságok, bejelentkezési vezérlőelemek, illetve jelszóparaméterek rendszerszintű alapértelmezéseinek létrehozása
- A rendszeradminisztrátor által a felhasználói fiókok létrehozásakor a felhasználók megbízhatóságát tükröző hitelesítési profilok beállítása
- Biztonsági attribútumok, SL és TL címkék eszközökhöz rendelése, például terminálok, nyomtatók, cserélhető lemezmeghajtók, illetve mágneses szalagmeghajtók esetében
- Biztonsági kapcsolók, címkék, jogosultságok és felhatalmazáshalmazok fájlokhoz rendelése
- A rendszer meghibásodása esetén a rendszer megbízható állapotba helyreállítása

Megfigyelési rendszer kezelése:

A megfigyelési parancsokhoz csak az **AUDITSYS** felhatalmazással rendelkező felhasználók férhetnek hozzá. További információkat az **audit**, az **auditselect** és az **auditpr** parancsok leírása tartalmaz.

Az alábbi példa bemutatja:

1. A nyomkövetési naplófájlokhoz használatos fájlrendszer létrehozásának módját
2. A megfigyelési rendszer indításának módját
3. Bizonyos rekordok előállításának módját
4. A különböző típusú rekordok lekéréséhez a nyomkövetési napló értelmezésének módját.

FSADMIN felhatalmazással rendelkező felhasználóként futtassa az alábbi parancsokat:

```
/usr/sbin/crfs -v jfs -g rootvg -m
/audit -a size=32M -A yes
mount /audit
```

A **/tbin/auctlmod -e** parancs segítségével adja hozzá a következő bejegyzést az /etc/security/audit/config fájlhoz:

```
felhasználónév = ALL
```

A *felhasználónév* érték helyett adjon meg egy tényleges felhasználót, aki be tud jelentkezni a rendszerre.

ISSO felhasználóként hozza létre a /tmp/top_secret nevű fájlt, majd módosítsa a fájl SL címkéjét **TS ALL** értékre.

```
touch /tmp/top_secret
/usr/sbin/settxattr -f sl= "TS ALL"
/tmp/top_secret
```

AUDITSYS felhatalmazással rendelkező felhasználóként futtassa az alábbi parancsot:

```
/usr/sbin/audit start
```

Ezzel a megfigyelési rendszer beállításra került, és rögzíti a *felhasználónév* által azonosított felhasználó tevékenységeit, amikor a felhasználó a rendszerre bejelentkezik.

Jelentkezzen be az /etc/security/audit/config fájlban a *felhasználónév* által meghatározott felhasználóként, majd futtassa a következő parancsokat:

```
ls -l /tmp/top_secret
exit
```

AUDITSYS felhatalmazással rendelkező felhasználóként futtassa az alábbi parancsot:

```
audit shutdown
$ /usr/sbin/auditselect -e "mac_fail==WILDCARD" /audit/trail | \
/usr/sbin/auditpr -v -APSV > /tmp/audit_trail-mac_failure
```

Vizsgálja meg a /tmp/audit_trail-mac_failure fájlba átirányított nyomkövetési naplót, majd keresse meg a **mac_fail** értéket. Az auditselect - a módosításokat követően - az alábbi beállításokat fogadja el:

- **subj_sl**
- **obj_sl**
- **mac_fail**
- **mac_pass**
- **mic_fail**
- **mic_pass**
- **priv_fail**
- **priv_pass**
- **auth_pass**
- **fsf_fail**
- **fsf_pass**

A beállítások megegyező értékként a **WILDCARD** értéket használják.

Objektum- és folyamatcímkék kezelése:

Minden fájlrendszer-objektum és rendszerfolyamat rendelkezik társított címkével.

A normál fájlok kivételével az összes fájlrendszer-objektum rendelkezik egy érzékenységi címke tartománnyal, illetve egy integritás címkével. A folyamatok az érzékenységi és integritás címkék esetében egyaránt tartománnyal rendelkeznek. A tartományok mellett a folyamatok rendelkeznek tényleges SL és TL címkével is. Ez a címke jelöli, hogy a folyamat jelenleg milyen SL és TL címkén fut. A címkéket a **lstxattr** parancs segítségével jelenítheti meg. A címkéket a fájlrendszer-objektumokon és folyamatokon a **settxattr** parancs segítségével állíthatja be.

Hálózati biztonság kezelése:

Az AIX Trusted Network megköveteli, hogy az ISSO számos táblát meghatározzon. A táblák az /etc/security könyvtárban kerülnek tárolásra. A bináris változat a **tninit** parancs segítségével állítható elő, majd tölthető be a kernelbe.

A hoszt és hálózati csatoló szabályok határozzák meg, hogy a rendszer milyen módon kezeli a bejövő és kimenő hálózati csomagokat. A hosztszabályok adott hosztokra vonatkoznak. A hálózati csatoló szabályok azokra a csatolókra vonatkoznak, amelyeken keresztül a hosztok a hálózathoz csatlakoznak. Ha egy hosztszabály és egy csatolószabály között ütközés áll fenn, akkor a hosztszabály élvez elsőbbséget.

Szabályok a **netrule** parancs segítségével vehetők fel, módosíthatók, illetve kérdezhetők le. Általánosságban a szabályok a felhasznált protokollokra, a szabályok céljaként megadott címtartományokra (hosztokra és portokra egyaránt), illetve a csomagokhoz hozzárendelendő SL címkékre vonatkoznak. További információkat a **netrule** parancs leírása tartalmaz.

Az AIX Trusted Network a **tninit** parancs segítségével inicializálható. A parancs segítségével ezen kívül a szabályok bináris formátumban elmenthetők és szöveges formátumban megjeleníthetők.

Biztonság konfigurálható szolgáltatásai:

A konfigurálható szolgáltatások beállításai a rendszerbetöltési szekvencia során jelennek meg.

A konfigurálható beállításokat az ODM tárolja. A beállítások a **getsecconf** parancs segítségével jeleníthetők meg, illetve az ISSO felhasználó a beállításokat a **setsecconf** parancs segítségével módosíthatja.

Címkék kezelése:

Az ISSO felhasználók az /etc/security/enc/LabelEncodings fájl módosításával kódolásokat vehetnek fel, módosíthatnak, illetve törölhetnek. Az /etc/security/enc/LabelEncodings fájl meghatározza, hogy a felhasználó által olvasható nevek milyen módon kerülnek leképezésre a rendszer érzékenységi címkéinek bináris ábrázolására.

Megjegyzés: Az érzékenységi címke kódolási fájl futó rendszeren történő módosítása érvénytelen címkéket eredményezhet, ha nem kellő körültekintéssel jár el. Mivel az objektumok címkéje egyedülálló szavakat és megszorítással rendelkező szókapcsolatokat egyaránt tartalmazhat, a szókapcsolatokat érintő megszorítások módosítása, felvétele és törlése érvénytelen címkékeket okozhat.

Az /etc/security/enc/LabelEncodings fájl fordítását bináris formátumúra az **l_init** függvénytárrutin végzi, a bináris formátumú adatok táblákban kerülnek tárolásra. A belső bináris kódolásról, illetve a kódolásra az SL címkék, nyomtató fejlécoldalak, illetve engedélyek ezen táblák segítségével kerülnek átalakításra.

A címkekezelés megvalósításához a Trusted AIX a MITRE Compartmented Mode Workstation Labeling szoftvert használja. A szabványos címkekódolási formátum magyarázatát a Compartmented Mode Workstation Labeling: Encodings Format, DDS-2600-6216-93 (MTR 10649 1. módosítás), 1993 szeptemberi kiadvány tartalmazza.

A szabványos címkekódolási formátum az integritás és érzékenységi címkéket egyező módon kezeli, az /etc/security/enc/LabelEncodings fájl **Sensitivity Labels** szakaszában meghatározott módon.

A Trusted AIX ezen kívül támogatja a nem kötelező **Integrity Labels** szakasz használatát, amely lehetővé teszi az integritás és érzékenységi címkék eltérő meghatározását.

Particionált könyvtárak kezelése:

A szokásos felhasználói folyamatok szempontjából a particionált könyvtárak normális könyvtárnak tűnnek és azokkal egyező módon viselkednek. Azonban a particionált könyvtárak esetében a különböző SL címkékkel rendelkező folyamatoknak ugyanabban a könyvtárban eltérő tartalom jelenik meg.

Ha például a **SECRET** biztonsági címkével futó folyamat egy particionált könyvtárban létrehoz egy **foo** nevű fájlt, akkor a második, TOP SECRET biztonsági címkével futó folyamat a könyvtárban a **foo** nevű fájlt nem látja, illetve a fájlhoz nem tud hozzáférni. Ezen kívül a második folyamat létrehozhat egy saját **foo** fájlt is, anélkül, hogy az első **foo** fájllal ütközés lépne fel.

Ezt a rendszer rejtett alkönyvtárak használatával oldja meg. Minden egyes egyedi SL címkére vonatkozóan, amellyel folyamatok a particionált könyvtárhoz hozzáférnek, létezik egy particionált könyvtár. Amikor egy folyamat a particionált könyvtárhoz hozzáfér, akkor a rendszer a folyamatot automatikusan a rejtett alkönyvtárba irányítja. A fenti példában a két **foo** fájl valójában különböző alkönyvtárakban található, bár a felhasználó számára úgy tűnik, mintha ugyanabban a könyvtárban lennének.

A particionált könyvtárakkal kapcsolatosan további információkat a “Particionált könyvtárak” oldalszám: 413 tartalmaz.

Az EAv2 fájlrendszert használó JFS2 fájlrendszer esetében a particionált könyvtárak használata támogatott.

Particionált könyvtár létrehozása:

A particionált könyvtárak létrehozásakor az alapértelmezett SL tartomány a rendszer alacsony SL és a rendszer magas SL közötti tartomány. Particionált könyvtár elérésekor a kernel automatikusan létrehoz egy, a címkére jellemző alkönyvtárat (kivéve, ha ilyen már létezik), majd a felhasználói folyamatokat ebbe az alkönyvtárba irányítja át.

Particionált könyvtár a **pdmkdir** paranccsal hozható létre. A **pdmkdir** parancs a DAC, MAC és MIC megszorítások felülbírálásához **aix.mls.pdir.create** felhatalmazást igényel. Az üres particionált könyvtárak a **pdrmdir** parancs segítségével távolíthatók el.

Particionált könyvtárak és alkönyvtárak

A particionált könyvtárak címkére jellemző alkönyvtárai particionált alkönyvtárak. Amikor egy folyamat (az **mkdir** parancs segítségével) particionált alkönyvtár alatt egy alkönyvtárt hoz létre, akkor az alkönyvtár particionált al-alkönyvtár lesz.

A particionált alkönyvtár létrehozásakor örökli a szülő particionált könyvtár biztonsági attribútumait, kivéve a minimális és maximális SL címkéket. A minimális és maximális SL címkéket a rendszer úgy állítja be, hogy azok a particionált alkönyvtárhoz először hozzáférő virtuális módú folyamat tényleges SL címkéi legyenek.

A Trusted AIX négyféle különböző könyvtártípust ismer fel:
- normális könyvtár (dir)
- particionált könyvtár (pdir)
- particionált alkönyvtár (psdir)
- particionált al-alkönyvtár (pssdir)

Virtuális és valós mód:

Két különböző particionált könyvtár-hozzáférési mód létezik: virtuális és valós mód.

Virtuális módban a particionált könyvtárat elérő folyamat csak a címkére jellemző particionált alkönyvtár tartalmát láthatja. A particionált könyvtár virtuális módban futó folyamat számára sosem látható. A particionált könyvtár a valós módban futó folyamat számára látható. A valós módban futó folyamatok a particionált könyvtárak és alkönyvtárak valós tartalmát láthatják. Valós módú folyamatok esetén a rendszer nem hajt végre átirányítást.

A folyamatok alapértelmezésben virtuális módban futnak. A valós módot csak fájlrendszer-adminisztrációs célokra szánták. A **pdmode** paranccsal a parancsokat az aktuális folyamat parancsértelmezőjétől eltérő módban futtathatja vagy átválthat egy más módban lévő parancsértelmezőre.

A valós módú folyamat láthatja és kezelheti a particionált könyvtárakat és alkönyvtárakat. Ezt a hozzáférési és kezelési típust körültekintően kell végrehajtani. Ha például egy valós módú folyamat létrehoz vagy áthelyez egy particionált könyvtárat, akkor a könyvtár sosem lesz látható a virtuális módban futó folyamatok számára.

A particionált könyvtár a virtuális módú folyamat számára ugyanúgy néz ki, mint a normál könyvtár, azonban korlátozások érvényesek rá.

Hierarchia:

A particionált könyvtárak és alkönyvtárak hierarchiát alkotnak.

A particionált könyvtárak és alkönyvtárak alkotta hierarchiát az alábbi szabályok vezérlik:
- A könyvtáraknak az alábbi négy típusból az egyikkel kell rendelkezniük:
  - normál könyvtár
  - particionált könyvtár
  - particionált alkönyvtár
  - particionált al-alkönyvtár
- Ugyanaz a könyvtár egyidejűleg több típussal nem rendelkezhet
- A particionált alkönyvtárak szülőjének particionált könyvtárnak kell lennie
- A particionált alkönyvtárak alkönyvtárának particionált al-alkönyvtárnak kell lennie
- A particionált al-alkönyvtárak szülőjének particionált alkönyvtárnak kell lennie

A fenti szabályok megsértése érvénytelen particionált könyvtárfát, illetve meghatározhatatlan viselkedésű, inkonzisztens fájlrendszert eredményez.

Fájlrendszerek felépítése:

A particionált könyvtárak és alkönyvtárak lehetnek felépítési pontok, de a particionált al-alkönyvtárak nem. Hasonlóképpen, a felépíteni kívánt fájlrendszer gyökere csak particionált könyvtár vagy alkönyvtár lehet, particionált al-alkönyvtár nem.

Könyvtárak létrehozása és törlése:

Amikor egy virtuális módú folyamat particionált al-alkönyvtárban fut, akkor az **mkdir** parancs normál könyvtárt hoz létre. Ha ugyanaz a folyamat egy particionált alkönyvtárban van és végrehajt egy **mkdir** parancsot, akkor automatikusan particionált alkönyvtár kerül létrehozásra. A MAC, MIC, és DAC megszorítások figyelembevételével tetszőleges üres könyvtár törölhető.

Könyvtárak áthelyezése:

A könyvtárak áthelyezésekor a MAC, MIC és DAC megszorítások érvényesülnek.

A normális könyvtárak tetszőleges helyre áthelyezhetők. Ha a könyvtár új szülőkönyvtára particionált alkönyvtár, akkor az áthelyezett normális könyvtár particionált alkönyvtár lesz. Ellenkező esetben továbbra is normális könyvtár marad. Ha a könyvtár új szülőkönyvtára particionált könyvtár, és a neve ütközik egy potenciális particionált alkönyvtár nevével, akkor a potenciális későbbi virtuális módú folyamatátirányítás az adott particionált alkönyvtárba meghiúsul.

A particionált könyvtárak áthelyezhetők másik normál könyvtárba; ezek az áthelyezés után továbbra is particionált könyvtárak lesznek. Az egymásba ágyazott particionált könyvtárakat a Trusted AIX nem támogatja, mivel ezek további előnyöket nem biztosítanak.

A particionált alkönyvtárak csak particionált könyvtárba helyezhetők át, és az áthelyezés után is particionált alkönyvtárak maradnak. A particionált alkönyvtárak normális könyvtárba, particionált alkönyvtárba, illetve particionált al-alkönyvtárba nem helyezhetők át.

A particionált al-alkönyvtárak tetszőleges helyre áthelyezhetők. Ha a könyvtár új szülőkönyvtára normális könyvtár, particionált könyvtár vagy particionált al-alkönyvtár, akkor a particionált al-alkönyvtárból normális könyvtár lesz. Egyébként particionált al-alkönyvtár marad.

42. táblázat: Könyvtáráthelyezés összefoglalása

Áthelyezni kívánt könyvtár típusa	**Normál könyvtárba**	**Particionált könyvtárba**	**Particionált alkönyvtárba**	**Particionált al-alkönyvtárba**
Normális	Engedélyezett. Normál könyvtár marad	Engedélyezett[1]. Normál könyvtár marad.	Engedélyezett[1]. Al-alkönyvtár lesz.	Engedélyezett. Normál könyvtár marad.
Particionált	Engedélyezett. Particionált könyvtár marad.	Engedélyezett[1]. Particionált könyvtár marad.	Nem engedélyezett.	Engedélyezett. Particionált könyvtár marad.
Particionált alkönyvtár	Nem engedélyezett.	Engedélyezett. Particionált alkönyvtár marad.	Nem engedélyezett.	Nem engedélyezett.
Particionált al-alkönyvtár	Engedélyezett. Normál könyvtár lesz.	Engedélyezett. Normál könyvtár lesz.	Engedélyezett. Al-alkönyvtár marad.	Engedélyezett. Normál könyvtár lesz.

[1] Ha a név ütközik egy potenciális (jelenleg nem létező) particionált alkönyvtár nevével, akkor az esetleges későbbi virtuális módú folyamatátirányítás az adott particionált alkönyvtárba meghiúsul.

Könyvtártípus módosítása:

A normál könyvtárak a **pdset** parancs segítségével módosíthatók particionált típusúra. A particionált könyvtárak normális típusúra módosítására parancs nem létezik.

Inode számok lecserélése:

Particionált alkönyvtár elérésekor ha az alkönyvtár inode számára vagy a szülő particionált könyvtárának (..) inode számára van szükség, akkor a szülő particionált könyvtárának az inode száma vagy szülő particionált könyvtára szülőjének az inode száma kerül visszaadásra, értelemszerűen. Ha egy particionált al-alkönyvtár elérésekor szükség van a particionált al-alkönyvtár szülő könyvtárának (..) inode számára, akkor a rendszer a szülő szülőjének particionált könyvtár inode számát adja vissza.

Particionált könyvtár parancsok:

A particionált könyvtárak esetében az alábbi parancsok használhatók.

pdmkdir
Particionált könyvtárak létrehozása

pdrmdir
Particionált könyvtárak és alkönyvtárak eltávolítása

pdlink Fájlok csatolása különböző particionált alkönyvtárak között

pdset Könyvtárak beállítása particionált könyvtárrá

pdmode
Az aktuális könyvtár hozzáférési módjának lekérdezése

Parancs futtatása a megadott könyvtár-hozzáférési módban

A particionált könyvtárrá átalakított szokásos könyvtár visszaalakítható szokásos könyvtárrá.

Rendszerbiztonság áttekintése:

Az ISSO felelőssége a rendszer biztonsági állapotának áttekintése. A rendszerbiztonság-áttekintést a telepítés után azonnal, a rendszerintegritás esetleges veszélyeztetésekor, valamint rendszeres időközönként is végre kell hajtani.

A rendszerintegritás-adatbáziskönyvtár, amely az /etc/security/tsd/tsd.dat fájlban kerül tárolásra, a fájlrendszer-objektumok biztonsággal kapcsolatos információit tartalmazzák, mint például a kritikus parancsok és rendszereszközök. Ezt az adatbázist frissíteni kell új eszköz hozzáadásakor és a fájlok biztonsági információinak módosításakor. További információkért tekintse meg a **trustchk** parancsot.

A **trustchk** parancs összehasonlítja a fájl, könyvtár vagy eszköz aktuális biztonsági beállításait a rendszerintegritási adatbázis megfelelő bejegyzésével és kijavítja a biztonsági attribútum inkonzisztenciáit. A **trustchk** parancsot csak az ISSO felhatalmazással rendelkező felhasználó futtathatja.

TTY kezelés:

Az eszközökhöz tartozó minimális SL, maximális SL és tty a ttys adatbázis /etc/login.cfg fájljában van megadva. További információkért tekintse meg a **chsec** parancsot.

A TTY porton keresztül bejelentkező felhasználó hatályos SL-jének a porthoz a fájlban megadott tartományban kell lennie. Ha nem NOTL TL van megadva a TTY porthoz, akkor a felhasználó hatályos TL-jének meg kell egyeznie a megadott TL-lel.

Felhasználói engedélyek kezelése:

Minden felhasználónak, az ISSO, SA és SO felhasználót is beleértve, rendelkeznie kell címkével a rendszerre való bejelentkezéshez. A felhasználói jogosultság az /etc/security/user fájlban, a felhasználói szakasz részeként adható meg. Az **minsl**, **maxsl**, **defsl**, **mintl**, **maxtl** és **deftl** attribútum megadja a felhasználó számára a minimális, maximális

és alapértelmezett SL-t, illetve a minimális, maximális és alapértelmezett TL-t, értelemszerűen. Ha ezek az attribútumok meg vannak adva a felhasználó szakaszában, akkor a fájl alapértelmezett szakaszában megadott értékek vannak a felhasználóhoz rendelve.

Csak ISSO felhasználó módosíthatja a biztonsági engedély adatbázist. A felhasználó jogosultsága az **lsuser** és **lssec** paranccsal jeleníthető meg, illetve a **chuser** és **chsec** paranccsal módosítható.

Az alapértelmezett SL értékkel szemben dominálnia kell a maximális SL értéknek, és az alapértelmezett SL értéknek dominálni kell a minimális SL-lel szemben. Ehhez hasonlóan az alapértelmezett TL értékkel szemben dominálni kell a maximális TL értéknek, és az alapértelmezett TL értéknek dominálni kell a minimális TL-lel szemben.

Megjegyzés: Ahhoz, hogy a felhasználó sikeresen bejelentkezzen a rendszerre, a fenti relációnak igaznak kell lennie.

Rendszerfelügyelet rendszeradminisztrátorok számára:

Az SA felhasználók elsősorban a rendszeradminisztráció biztonsággal nem kapcsolatos részeiért felelősek.

Az SA felhasználók felelősek - többek között - az alábbiakért:

- Felhasználói fiókok felvétele, eltávolítása és karbantartása
- Az ISSO felhasználóval közösen a rendszerszoftver és a fájlrendszerek belső integritásának biztosítása
- Fájlrendszerek létrehozása és karbantartása. Ebbe beletartozik - többek között - a lemezszerkezet tervezése, a lemezek particionálása és a lemezpartíciók méretének módosítása, a lapozási területhez, illetve a rendszer- és felhasználói könyvtárakhoz terület lefoglalása, a fájlrendszer használatának megfigyelése, a rossz lemezblokkok felismerése és kezelése, illetve - a fájlok és fájlrendszerek áthelyezésén, törlésén, archiválásán vagy tömörítésén keresztül - a fájlrendszerterület kezelése.
- A rendszerproblémák azonosítása és jelentése a hibaadatok elemzésével, illetve a rendszer összetevőinek (például a fájlrendszerek, a rendszermemória és az eszközök) tesztelésével.

Felhasználói fiókok kezelése:

Az SA felhasználó felelős az új felhasználók rendszerhez adásáért. Az ISSO felhasználó felelős annak engedélyezéséért, hogy az új felhasználók bejelentkezhessenek és végrehajthassák a parancsokat a rendszeren.

Felhatalmazások felhasználói fiókokhoz adásával kapcsolatos információkért tekintse meg a Rendszer kezelése az információs rendszer adatvédelmi megbízottak számára részt.

Ha az SA felhasználó hozzáadott egy felhasználót a rendszerhez, akkor az ISSO felhasználót értesíteni kell, hogy a kezdeti jelszó beállítható legyen az új felhasználó rendszerhozzáférésének engedélyezéséhez.

Ha meg lett határozva, hogy a felhasználó már nem férhet hozzá a rendszerhez, akkor a felhasználót azonnal el kell távolítani. A felhasználó eltávolítását csak SA felhasználó végezheti. A rendszerről eltávolított felhasználó felhasználói azonosítója nem használható fel újra, hacsak az eredeti felhasználó vissza nem kapja, és csak a felhasználó újbóli példányosításakor.

A felhasználói fiókok létrehozásával és módosításával kapcsolatos információkért tekintse meg a **mkuser**, **rmuser**, **chuser** és **pwadm** parancsot.

Nyomtatók kezelése:

Miután egy nyomtató megfelelően telepítésre került, a rendszerhez hozzáadásra kerül az SA és SO felhasználók egyesített tevékenységén keresztül kerül. Az SO felhasználó a nyomtatót hozzáadja a rendszerhez, majd az SA felhasználó létrehozza a nyomtató SL tartományát. Az ISSO felhasználók mindkét feladat végrehajtásához rendelkeznek jogosultsággal.

A nyomtató SL tartományát nem szabad addig beállítani, amíg a nyomtató a rendszerhez hozzáadásra nem került. A nyomtatók az **smit** parancs segítségével kezelhetők.

Megjegyzés: A PostScript és ASCII fájlok címkéket használó nyomtatása csak PostScript nyomtatók esetében támogatott.

A nyomtató MAC hozzáférését a fájlt nyomtató folyamat SL címkéje határozza meg. Az SL megjelenik a fejlécoldalon, a fejlécen/láblécen, illetve a befejező oldalakon. Az **lp** parancsot használó folyamatoknak MAC, MIC, és DAC hozzáféréssel kell rendelkezniük a nyomtatás alatt álló fájlra vonatkozóan. Ellenkező esetben az **lp** parancs a nyomtatási kérést nem állítja elő.

Ha egy nyomtató a rendszerről eltávolításra kerül, akkor a nyomtatóprofilt ajánlott azonnal törölni a rendszerről. Ezt csak SO felhatalmazással rendelkező felhasználók hajthatják végre.

Fájlrendszerek kezelése:

A fájlrendszerek könyvtárakból, adatfájlokból, végrehajtható fájlokból és speciális fájlokból állnak. A fájlrendszerek különböző adattároló-eszközökön helyezkedhetnek el, például merev- és hajlékonylemezeken.

Bár csak a fájlrendszerek létrehozását és karbantartását csak az SA felhasználók végezhetik el, fájlrendszereket az SA és SO felhasználók egyaránt felépíthetnek és lebonthatnak.

Fájlrendszerek ellenőrzése az fsck paranccsal:

A fájlrendszerek belső integritását tanácsos rendszeres időközönként az **fsck** parancs segítségével ellenőrizni. Az **fsck** parancsot lebontott fájlrendszereken kell futtatni. Az **fsck** parancsot csak SA felhasználók futtathatják.

Alapértelmezésben az **fsck** parancs interaktív módon fut, és a felhasználót megkérdezi, hogy árva fájl vagy könyvtár felfedezésekor milyen tevékenységet hajtson végre. A felhasználó dönthet a fájl törlése mellett, illetve megkísérelheti a fájl helyreállítását. Ha a felhasználó a fájl helyreállítása mellett dönt, akkor az **fsck** parancs kísérletet tesz a fájl eltárolására a /lost+found könyvtárban.

Miután az **fsck** parancs befejeződött és a helyreállított fájlok eltárolásra kerültek a /lost+found könyvtárban, egy ISSO felhasználónak át kell tekintenie a fájlokat és meg kell határoznia a fájlok biztonsági szintjét. Tanácsos a /lost+found könyvtárhoz **SYSTEM_HIGH** SL címkét rendelni, és ezáltal megakadályozni, hogy a normális felhasználók a helyreállított fájlokhoz hozzáférhessenek.

További információkat az **fsck** parancs leírása tartalmaz.

Rendszer kezelése a rendszermegbízottak számára:

Az S0 felhasználók elsődlegesen a rendszeradminisztráció biztonsággal kapcsolatos szempontjaiért felelősek.

Fájlrendszerek kezelése:

A fájlrendszer kezeléséért felelős rendszermegbízottak

Támogatott fájlrendszerek:

A Trusted AIX támogatja az összes lemezalapú fájlrendszert.

A JFS2 kivételével minden fájlrendszer támogatott Trusted AIX rendszeren egyszintű fájlrendszerként. Ezek a fájlrendszerek Trusted AIX rendszeren felépíthetők, automatikusan fogadják a címkéket és más biztonsági attribútumokat, és megcélozzák a Trusted AIX által betartatott biztonsági mechanizmusokat. Egyszintű fájlrendszeren minden fájlrendszer-objektum azonos biztonsági attribútummal rendelkezik. Ezeket a biztonsági attribútumokat a felépítési ponttól öröklik.

A JFS2 a Trusted AIX rendszeren többszintű fájlrendszerként kerül megvalósításra. A többszintű fájlrendszer minden fájlobjektuma saját biztonsági attribútumokkal rendelkezik (biztonsági címkék). A JFS2 könyvtár például független minimális és maximális SL-lel rendelkezik.

Egyszintű fájlrendszeren a felépítési pont minimális és maximális SL-jei azonosak és a felépítési pont alatt lévő minden könyvtárnak illetve fájlnak szintén egyenlő kell lennie ezekkel az SL-lekkel.

Fájlrendszerek felépítése és lebontása:

Az SO felhasználó (**aix.fs.manage.mount** felhatalmazással) felépíthet vagy lebonthat fájlrendszert. A **mount** parancs eszközspecifikus fájlnevet és a felépítési könyvtárat paraméterként használja.

Több szintű JFS2 fájlrendszer felépítése esetén a felépítési könyvtárhoz hozzárendelésre kerül a fájlrendszer root címkéje. Többszintű fájlrendszeren minden fájl saját érzékenységi és integritási címkével rendelkezik. A fájl módosítása esetén a címkéje megfelelően módosul.

Nyomtatók kezelése:

Az SO felhasználó az **lpadmin** parancs segítségével nyomtatókat vehet fel, távolíthat el és módosíthat, valamint más típusú vezérlést gyakorolhat a nyomtató alrendszeren. Az SA felhasználó az **lpadmin** parancs segítségével egy nyomtatóhoz adhat érzékenységi címkéket, vagy módosíthatja a meglévőket, illetve az enable és disable parancs segítségével nyomtatókat engedélyezhet vagy tilthat le.

Nyomtatóalrendszer:

A nyomtatóalrendszer a nyomtatóművelettel kapcsolatos feladatokat hajt végre.

A nyomtatóalrendszer-feladatok a következők:
- Nyomtatók és azok attribútumainak felügyelete
- Felhasználói nyomtatási feladatok fogadása, tárolása és ütemezése
- nyomtatási feladatok ütemezése több nyomtatóhoz
- A nyomtatóval érintkező programok indítása
- Nyomtatók és nyomtatási feladatok állapotának nyomkövetése
- Problémák jelentése felmerülésük estén
- Felhasználói nyomtatási feladatok korlátozása a nyomtatók SL tartományába esőkre
- felhasználói nyomtatási feladatok korlátozása elküldéskor
- Nyomtatótámogatási fájlok és könyvtárak elérésének korlátozása
- Nyomtatókimenet megfelelő címkézése

Nyomtatóbiztonsági szolgáltatások:

A nyomtatóalrendszer az Trusted AIX rendszeren módosítva lett számos biztonsági szolgáltatás egyesítése érdekében.

A nyomtatóalrendszer az **lp** rendszerazonosító által birtokolt védett alrendszer. Ez megakadályozza, hogy a normál felhasználók elérjék a nyomtatótámogatási fájlokat és könyvtárakat, a felhasználó sajátjától különböző elküldött nyomtatási feladatokat és a nyomtatóeszköz speciális fájljait.

A nyomtatóalrendszer ellenőrzi, hogy a felhasználó elküldött nyomtatási feladata a nyomtató SL tartományába esik-e. Ez az ellenőrzés akkor kerül végrehajtásra, amikor a felhasználó az **lp** paranccsal elküld egy nyomtatási feladatot és mielőtt az elküldött feladatot az **lpsched** démon kinyomtatná. Az adminisztrátornak ismernie kell a nyomtatóalrendszer biztonsági ellenőrzéseit abban az eseten, ha egy felhasználó nyomtatási feladata visszautasításra kerül.

A fejlécoldalak minden nyomtatási feladathoz kinyomtatásra kerülnek. A fejlécoldal tartalmazza a nyomtatási feladat felhasználó által olvasható SL-jét. A fejlécoldal a nyomtatási feladatok elején és hátulján jelenik meg. Minden

felhasználó nyomtathat fejléc nélkül, de ez egy megfigyelhető tevékenység. Mindig ellenőrizni kell, hogy az oldalon lévő fejléc és lábléc címke helyes, és a fejlécoldalon lévő címkék dominálnak.

Megjegyzés: A sornyomtató-adminisztrátornak minden nyomtatóhoz ki kell alakítani a címketartomány. Címke nyomtatóhoz rendelése érdekében futtassa a következő parancsot:

lpadmin -d *nyomtatónév* **-J***címke* **-L***címke* Ez biztosítja, hogy csak a megadott *címkével* rendelkező információkat lehessen kinyomtatni a nyomtatón.

Nyomtatóparancs összefoglalása:

A nyomtatóalrendszer-parancsokat minden felhasználó futtathatja. Azonban néhány nyomtatóalrendszer-parancsot csak SO, SA vagy ISSO felhasználó futtathat.

A következő táblázat az összes felhasználó által futtatható nyomtatóalrendszer-parancsokat sorolja fel:

lp Fájlt küld a nyomtatóra

lpstat Állapotjelentést a nyomtatóalrendszerről

A nyomtatóalrendszer-adminisztrációs parancsok SO felhatalmazást igényelnek, ez alól kivétel, hogy az SA vagy ISSO felhatalmazással rendelkező felhasználó futtathatja az **lpadmin** parancsot a nyomtató címtartományának megadásához, valamint az **lpstat** parancsot a nyomtató- és feladatkérés SL-ek megjelenítéséhez. A következő táblázat a nyomtatóalrendszer-adminisztrációs parancsokat tartalmazza:

accept Engedélyez egy feladatot a nyomtatón

cancel Visszavonja egy fájl nyomtatási kérését

disable Leállít egy nyomtatót

enable Aktivál egy nyomtatót

lpadmin
Beállítja vagy módosítja a nyomtatókonfigurációt

lpfilter Beállít vagy módosít egy nyomtatószűrőt

lpforms
Beállít vagy módosít egy nyomtatóformátumot

lpmove
Áthelyezi a nyomtatási kéréseket

lpsched
Kinyomtat egy kérést

lpshut Leállítja a nyomtatási szolgáltatást

lpusers
Beállítja vagy módosítja a nyomtatási prioritást

reject Letiltja a feladatokat egy nyomtatón

Parancssori nyomtatókezelés:

Az **accept**, **enable**, **disable**, **lpstat** és **lp** parancs segítségével a nyomtató parancssorból kezelhető.

Az **accept** parancs segítségével engedélyezheti a feladatok nyomtatóra küldését. Futtassa a következő parancsot annak engedélyezéséhez, hogy a *laser* nyomtató nyomtatási feladatokat fogadhasson:

```
/usr/sbin/accept laser
```

A *laser* által megadott nyomtató nyomtatási feladat kéréseket fogadhat. A nyomtatási feladatok azonban nem kerülnek kinyomtatásra, hacsak a nyomtató nem engedélyezett. Futtassa az enable parancsot a nyomtató engedélyezéséhez:

```
/usr/bin/enable laser
```

Az **enable** és **disable** parancs adminisztrátori parancs, és csak ISSO vagy SA felhatalmazással rendelkező felhasználó futtathatja.

A nyomtató megfelelő beállításának megerősítéséhez futtassa a következő **lpstat** parancsot:

```
lpstat -p laser -l
```

Ez a parancs a *laser* nyomtató hosszú állapotjelentését jeleníti meg. Ha az **lpstat** parancsot **-l** paraméter nélkül futtatja, akkor egy rövidebb állapotjelentés jelenik meg. Ha a felhasználó SA vagy ISSO felhatalmazással rendelkezik és a **-l** paraméter használatban van, akkor a nyomtató SL tartománya szintén jelentésre kerül.

A nyomtatási kérés állapotának meghatározása érdekében futtassa a következő **lpstat** parancsot:

```
lpstat -o
```

Ez a parancs megjeleníti az összes **lp** nyomtatási kérést. Ha a felhasználó SA vagy ISSO felhatalmazással rendelkezik, akkor a hatályos SL és az egyes kérések engedélyez kerül jelentésre.

A fájlnév kinyomtatásához futtassa a következő **lp** parancsot:

```
lp -d laser
fájlnév
```

Ellenkező esetben meg kell adni a nyomtatási feladat célját az **lp** parancs futtatásakor.

Ha az adminisztrátor beállított egy alapértelmezett nyomtatót, akkor a **-d** *destination_ptr* paraméter nem szükséges. Például a fájlnév fájl laser nevű nyomtatón való nyomtatásához írja be a következő **lp** parancsot:

```
lp fájlnév
```

Rendszerleállítás kezelése:

Az SO felhasználó le tudja állítani a rendszert a rendszer újraindításával vagy teljes leállításával.

Az SO felhasználó a következő parancsokat futtathatja a rendszer újraindításához vagy leállításához, illetve a rendszer kezdeti állapotának módosításához:

reboot Automatikusan újraindítja a rendszert

halt Leállítja az összes rendszerműveletet

leállítás
Leállítja az összes rendszerműveletet

init Módosítja a rendszer kezdeti állapotát

Fájlmentés és -visszaállítás:

A biztonsági mentések elősegítik a hardverhiba vagy a fájl véletlen törlése esetén fellépő adatvesztés megakadályozását. A biztonsági mentéseket rendszeres időközönként kell elvégezni, növekményes mentésekkel a teljes mentések között.

A **backup** és **restore** parancs a fájlmentések nevének, helyének, típusainak beállítására szolgáló paramétereket és egyéb paramétereket tartalmaz. Az **mksysb** parancs segítségével létrehozhatja a root kötetcsoport Trusted AIX telepítőhalmazát egy fájlban vagy egy betölthető szalagon. Ezek a parancsok az **smit** paranccsal futtathatók. A fájlrendszermentéseket megfelelően kell címkézni és tárolni egy biztonságos helyen.

Trusted AIX programozása

A rendszerbiztonság a Megbízható számítástechnikai alapkörnyezet (TCB) szoftverétől, hardverétől és firmware-jétől függ. Ez a teljes operációs rendszer kernelt, minden eszközillesztőt és System V STREAMS modult, kernelbővítményt és minden megbízható programot tartalmaz. A programok által használt fájlokat a rendszer a biztonsági döntésekben a TCB részének tekinti.

A megbízható szoftver létrehozása az alap rendszerbiztonsági elvek és szolgáltatások alapos ismeretét igényli. A UNIX alapú rendszerek majdnem minden biztonsági hiányának oka a nem megfelelően megírt megbízható szoftver. A Trusted AIX kernel biztonsági ellenőrzésekkel kiterjesztett biztonsági szolgáltatásokat használó alkalmazásokat írhat. A Trusted AIX rendszerhez írt alkalmazás érzékeny lehet a fájlokra és folyamatokra különböző biztonsági szinteken, és különféleképp működhet az alkalmazás által használt folyamat vagy fájl szintjétől függően. Az ilyen alkalmazás többszint kezelésére felkészített MLS alkalmazás.

A megbízható rendszer programozójának jártasnak kell lennie a Trusted AIX biztonsági szolgáltatásokban és ismernie kell az összes új Trusted AIX rendszerhívást, illetve biztonsággal kapcsolatos parancsot és könyvtárat. Ezek az információk megbízható szoftvert létrehozó és módosító programozók számára biztosítottak. Irányelveket, elveket és figyelmeztetéseket tartalmaznak a megbízható szoftver módosításával és létrehozásával kapcsolatban. Annak ellenére, hogy ezek az információk bevezető magyarázatokat biztosítanak néhány biztonsági elvhez és metódushoz, ajánlatos a megbízható rendszerprogramozók számára más biztonságos rendszerekkel kapcsolatos anyagokat is elolvasniuk.

Megbízható szoftverek alapelvei

A megbízható szoftverek létrehozásában és módosításában számos fontos elv játszik szerepet, többek között a megbízhatóság és a jogosultságok, a megbízható szoftvertervezés, a legkevesebb jogosultság, a programozási egyezmények és a TCB védelme.

Megbízhatóság és jogosultság:

A folyamat csak akkor lépheti át az alapvető biztonsági megszorításokat (MAC, MIC, DAC és más korlátozott műveletek), ha a folyamat megfelelő jogosultságokkal rendelkezik. A jogosultsággal vagy jogosultságokkal futó folyamatot privilegizált folyamatnak, a folyamat által futtatott programot pedig privilegizált (megbízható) programnak hívjuk.

A jogosultság kifejezés egy egyéni attribútumra utal, amely lehetővé teszi, hogy a folyamat biztonsággal kapcsolatos műveletet hajtson végre. A Trusted AIX adott biztonsági műveleteket azonosít és csoportosít, illetve adott jogosultságot társít az egyes műveletekhez. Ez ténylegesen eltávolítja a felettes felhasználó (vagy root) jogosultságot az alap rendszerről. A jogosultságok folyamatokhoz és végrehajtható fájlokhoz vannak társítva.

A programoknak a következő körülmények között megbízhatóknak kell lenniük:

- A program privilegizált folyamatként van beállítva vagy aként fog futni. Ez privilegizált folyamat által futtatandó programokra érvényes.
- A program másik megbízható programra támaszkodik a biztonsági döntések meghozásában. Az érzékeny adatbázist módosító programnak például megbízhatónak kell lennie, ha más programok az adatbázisban lévő adatokra támaszkodnak a biztonsági döntés meghozásához.

Fontos annak biztosítása, hogy a nem megbízható programok sose futhassanak privilegizált folyamatként. Számos lehetőség áll rendelkezésre annak megakadályozására, hogy a nem megbízható programok privilegizált folyamatként fussanak:

- Normális esetben ne engedélyezze, hogy a privilegizált folyamat nem megbízható programokat hajtson végre. Privilegizált parancsértelmező-szerű programokat futtató felhasználók esetén figyeljen arra, hogy ne futtathassanak nem megbízható programokat privilegizált parancsértelmező-szerű programban.
- Sose engedélyezzen belső, örökölt vagy felhatalmazott jogosultságokat nem megbízható végrehajtható fájlokhoz.

Az operációs rendszer kernel minden részének megbízhatónak kell lennie, az eszközillesztőket, STREAMS modulokat és kernelbővítményeket is beleértve. Az adatobjektumok, mint például a fájlok és fizikai eszközök, szintén megbízhatók, ha olyan információkat tartalmaznak, amelyekre megbízható program támaszkodik a biztonsági döntések meghozásához.

Megbízható szoftver tervezése:

Megbízható szoftver létrehozásának folyamata hasonló a kritikus szoftverösszetevőkéhez. A megbízható szoftver létrehozásának körültekintően megismert és dokumentált specifikációt, tervezést, megvalósítást, tesztelést és konfigurációvezérlési ciklust kell követnie.

A megbízható szoftvertervezés legfontosabb szempontjai: az alanyok és objektumok azonosítása és a pontos biztonsági tevékenységek meghatározása a megfelelő absztrakciós szinten. A legtöbb biztonsági irányelv az alanyokra, objektumokra és tevékenységekre vonatkozó korlátozás. Ha az alanyok jogosultságot kérnek objektumok olvasására, megváltoztatására vagy létrehozására, akkor a biztonsági irányelvek megfigyelik ezeket és kéréseket, majd elfogadják vagy visszautasítják azokat.

Alanyok

Az alanyt normális esetben egy felhasználói azonosító vagy csoportazonosító ábrázolja. Normális esetben a folyamat hatályos felhasználó és/vagy csoportazonosítója kerül felhasználásra erre a célra, azonban néhány esetben a valós felhasználó és/vagy csoportazonosító használata lehet megfelelő.

Objektumok

Az objektum azon adatok gyűjteménye, amelyek hozzáférését vezérelni kell. A legtöbb esetben az objektumok fájlok. Megbízható programok esetén általános a fájlban lévő logikailag különálló objektumok hozzáférésének vezérlése. Általában jobb gyakorlat az objektumok egyértelmű leképezése a fájlokra.

Néhány esetben az alany objektum is lehet. A folyamat például normális esetben alany. Amikor azonban egy folyamat megpróbál egy második folyamatot befolyásolni, akkor a második folyamatot a rendszer normális esetben objektumnak tekinti a művelet figyelembe vételével.

Kérések

A kérések a megbízható modul által egy alany helyett végrehajtott tevékenységek halmazai. Minden kérést tisztán kell azonosítani a kérés bemenetei, lehetséges kimenetei és eredményei alapján, a mellékhatásokat is beleértve. Az összes kérés pontos azonosítása fontos előzménye a biztonsági irányelvek meghatározásának.

Biztonsági irányelvek

A biztonsági irányelvek egyszerű utasításokat tartalmaznak, amelyek jelzik, ha a megadott objektumokat érintő kérések kerülnek végrehajtásra a megadott alanyok helyett. Az alanyokat, objektumokat és kéréseket körültekintően kell megadni, valamint a biztonsági irányelveknek tömörnek és magától érthetődőnek kell lenniük. Fontos a kérelmező alany azonosságának, valamint a megfigyelésbe bevont objektumok megadása.

Legkevesebb jogosultság:

A legkevesebb jogosultság alapelve kimondja, hogy a szoftvermodulok az elvégezni kívánt feladat végrehajtásához a minimális képességgel rendelkezzenek.

A legkevesebb jogosultság magában foglalja azt az elvet, hogy a megbízható programok önkéntesen korlátozzák saját érzékeny képességeiket, hogy ezáltal ezek a program lehető legkevesebb területén legyenek felhasználhatók. A legkevesebb jogosultság segít a szoftverhibákból és nem várt mellékhatásokból származó károk csökkentésében. Minden megbízható szoftver tervezése során ajánlott a legkevesebb jogosultság elvét figyelembe venni.

Jogosultság adása és eltávolítása:

A programok által használt egyik megbízható szoftveres módszer, hogy a program az összes jogosultságot igénylő műveletet a végrehajtás első szakaszában végrehajtja, majd a jogosultságról a működés hátralévő szakaszában lemond. Ezt a módszert hívják jogosultságbefogásnak.

A jogosultságok használatával kapcsolatosan ne feledkezzen meg az alábbiakról:

- Minden felhasználó folyamatának a végrehajtáskor kiosztásra kerül a maximális jogosultsághalmaz. A jogosultsághalmaz bármikor csökkenthető, de a nem privilegizált felhasználó a halmazt nem növelheti.
- A privilegizált műveletek végrehajtásakor a maximális halmaz jogosultságainak a hatályos halmazra növelése, illetve csökkentése a végrehajtó folyamat feladata.
- A folyamat jogosultságait a rendszer akkor módosítja, amikor a folyamatok ürestől eltérő belső jogosultsághalmazzal rendelkező végrehajtható fájlokat futtatnak. További információkat az **exec** parancs leírása tartalmaz.
- A folyamatok futtatásukkor is korlátozó jogosultsághalmazt kapnak. A megfelelő jogosultsággal a folyamatok a maximális halmazban meghatározott jogosultságokat megnövelhetik a korlátozó halmaz jogosultságaira.

Rövid MAC címke módosítások:

Ha egy folyamatnak meg kell változtatnia a MAC címkéjét a normál működési címkéről, akkor a címkemódosítás időtartamának a lehető legrövidebbnek kell lennie. Ez a függvénytár-rutinok használatával érhető el.

A függvénytár-rutinok használatával kapcsolatos információkért tekintse meg a “Megbízható AIX rendszerhívások” oldalszám: 480 részt.

Érzékeny fájlok rövid idejű megnyitása:

Az érzékeny fájl, mint például az árnyék jelszófájl, a rendszerbiztonságot veszélyeztető információkat tartalmaz. Ha az érzékeny fájl megnyitásra kerül olvasásra vagy írásra, akkor azt csak a szükséges ideig szabad nyitva tartani.

A fájlleíró **close-on-exec** attribútumát az **fcntl** rendszerhívással kell beállítani. Ez megakadályozza, hogy jogosulatlan folyamatok örököljék a megnyitott fájl leíróit az **exec** rendszerhíváson keresztül.

Érzékeny műveletek központosítása:

Az érzékeny műveletek olyan műveletek, amelyek jogosultságokat igényelnek. Ha egy nem privilegizált folyamat érzékeny műveletet hajt végre, akkor az veszélyeztetheti a rendszer biztonságát.

Az érzékeny műveleteket tanácsos a különálló modulok (szubrutinok vagy önálló programok) körére korlátozni. Azáltal, hogy a nagyobb programokat önálló programokra bontja, bizonyos programoknak kevesebb jogosultságra lesz szükségük (illetve előfordulhat, hogy egyáltalán nem lesz jogosultságra szükségük). Ezzel a módszerrel csökkenthető a rendszerbiztonság véletlen veszélyeztetésének lehetősége.

Hatályos root könyvtárak használata:

A program adott könyvtárfára korlátozható a program hatályos gyökérkönyvtárának beállításával a fa alapkönyvtárára (a **chroot** rendszerhívás segítségével) és a program munkakönyvtárának beállításával ugyanezen fában. Valójában ez egy legkevesebb jogosultságú mechanizmus, mivel korlátozza a fájlokat, még azokat is, amelyhez privilegizált folyamat férhet hozzá a fában. Ez különösen hatékony lehet, ha a szülő (megbízható) folyamat ilyen módon korlátozza a megbízható és nem megbízható leszármazott folyamatokat.

A gyökérkönyvtárak módosítása védelmet nyújt az új root fán kívül lévő fájlok számára, de ez potenciális biztonsági problémát jelent. A gyökérkönyvtár módosítása az új root fa biztonságát veszélyeztetheti, ha nem körültekintően történik. Ez akkor lép fel, ha a futási összekapcsoló és az új root fában lévő osztott objektumok meghamisíthatók. Ezt az eljárást körültekintően és korlátozottan kell használni.

Védett alrendszerek használata:

A védett alrendszerek integritásvédelmet biztosítanak speciális alrendszerekhez. Az alrendszer programok és/vagy adatfájlok gyűjteménye, amelyeket ugyanaz a felhasználói azonosító és/vagy csoportazonosító birtokol, mit amely a rendszer egy adott funkcióját valósítja meg.

Az alrendszer setuid vagy setgid programokat tartalmazhat. A védett alrendszer rendszerfelhasználói azonosítóval rendelkező alrendszer.

A rendszerfelhasználói azonosító 127-nél nem nagyobb értékkel rendelkező felhasználói azonosító. A felhasználók nem jelentkezhetnek be rendszerfelhasználói azonosítóval. A védett alrendszer használata jelentősen csökkenteheti a privilegizált folyamatok számát.

Minimális hozzáférési módok:

A megbízható programoknak (valójában az összes programnak) csak akkor ajánlott objektumokat írási/olvasási módban megnyitni, ha ez feltétlenül szükséges. Alapvetően tehát az objektumokat nem szabad írási és olvasási módban megnyitni akkor, ha az olvasási módban történő megnyitás is elegendő. A különösen érzékeny helyzetekben ajánlott, hogy a folyamat csak írási módban nyissa meg az objektumot azokon a különleges helyeken, ahol az írás szükséges.

Ezek az eljárások főleg akkor fontosak, ha a program egyéb folyamatokat hoz létre, hiszen a jogosultságok és egyéb általános képességek átadása (például kapcsolatok megnyitása érzékeny fájlok felé) a megbízható szoftvertervezés fontos szempontjai. A jogosultságok valamennyi megszorítást felülbírálhatják. A jogosultsággal rendelkező parancsok létrehozása rendkívüli körültekintést és pontos tervezést igényel.

Egyéb megbízható programozási egyezmények:

A Trusted AIX számos egyéb megbízható programozási egyezményt használ.

Redundancia:

A redundancia a biztonságos rendszerek számára hasznos eljárás. A biztonság ritkán tökéletes, és biztosítása többnyire abból áll, hogy a rendszerhez jogosulatlanul hozzáférő egyén útjába megfelelő számú akadályt helyezünk el.

A redundáns biztonsági ellenőrzések előnye, hogy amennyiben az egyik ellenőrzés meghibásodik vagy biztonsága sérül, akkor egyéb ellenőrzések biztosíthatnak védelmet. A redundáns ellenőrzések hátránya, hogy az átfogó biztonsági ellenőrzések egymástól elválasztásra vagy a rendszeren szétosztásra kerülnek. Ennek következtében, bár a redundáns ellenőrzések rendkívül hasznosak lehetnek, gondos tervezést, dokumentációt és karbantartást igényelnek.

Többszörös kernelellenőrzések elkerülése:

Kevés olyan eset létezik, amikor ajánlott, hogy a folyamat hajtson végre egy kernel által végrehajtható ellenőrzést. Nem ajánlott például, hogy a folyamatok beolvassák egy fájl MAC címkéjét, majd a MAC ellenőrzést saját maguk végezzék el. Amikor csak lehetséges, az ellenőrzést ajánlott magára a kernelre bízni.

Két fő oka van annak, hogy az ellenőrzéseket a kernel végezze el.

- A kernelműveletek a többi folyamat szempontjából elhanyagolhatók, míg a folyamatok által végzett ellenőrzések ténylegesen versenghetnek az egyéb folyamatokkal.
- Ami ennél is fontosabb, hogy a használt pontos algoritmusok a kernel újabb változataival párhuzamosan változhatnak. Az ilyen változások nehezen követhetők, ha az algoritmusok a végfelhasználói szoftver részét képezik.

Közvetlen jogosultságellenőrzés:

Nem ajánlott, hogy a programok maguk próbálják meg meghatározni, hogy jogosult folyamatként kerültek-e meghívásra (például hatályos vagy maximális jogosultságvektoruk megvizsgálásával). Ehelyett a programoknak ajánlott feltételezniük, hogy jogosultként kerültek meghívásra, amikor csak lehetséges.

Ha a program nem privilegizált folyamat, akkor a privilegizált rendszerhívás meghiúsul és a program végrehajthatja a megfelelő műveleteket. Általában nem hatékony biztonsági óvintézkedés, ha maga a program utasítja vissza bizonyos műveletek végrehajtását akkor, ha a művelet nem privilegizált. Ha a program privilegizált, akkor az ellenőrzés felesleges. Ha a program nem privilegizált, akkor a program több kárt okozhat, mint az egyéb nem privilegizált folyamatok.

Azonban az ellenőrzés hatékony segédeszközként használható a véletlen visszaélések ellen. Megjelenhet egy hibaüzenet, amely tájékoztatja a felhasználót, hogy bár a programot privilegizáltnak szánták, mégsem az.

Érzékeny képességek terjesztése:

Az érzékeny képességek a megbízható programok olyan képességei, amelyek veszélyeztethetnék a rendszer biztonságát akkor, ha a nem megbízható programoknak is biztosítva lennének.

Tanácsos óvatosan eljárni akkor, amikor egy megbízható program jogosultságait és általános képességeit egyéb programok felé a **fork** és **exec** rendszerhíváscsalád tagjain keresztül terjeszti. Az **exec** rendszerhívások a legfontosabbak, mivel ezek jogosultságokat két program között adnak át. A **fork** rendszerhívás új folyamatot hoz létre, de az új folyamatjogosultságok a szülő jogosultságokkal megegyezők. Az elsődleges veszély, hogy a végrehajtható programfájl nem feltétlenül megbízható, illetve előfordulhat, hogy egy megbízhatatlan program a programfájlt módosította. Az alábbiakat tanácsos szem előtt tartani:

- A megbízható programoknak ajánlott körültekintően eljárniuk, hogy nyitott kapcsolatokat utódfolyamatok objektumai (elsődleges fájlok) felé ne adjanak át, hacsak az utódban, illetve leszármazottaiban nem lehet megbízni, hogy azok a fájlhoz annak megnyitási módjában, megfelelően férnek hozzá. A legjobb ötlet talán, hogy a folyamat új kapcsolatot adjon át az olyan objektum felé, amelynek módjai korlátozottabbak az egyébként létezőknél.
- Az olyan megbízható folyamatoknak, amelyek tényleges gyökérkönyvtára az abszolút gyökérkönyvtártól eltérő, meg kell győződniük arról, hogy utódfolyamataik nem keverhetők össze. Amikor például egy utódprogram egy megbízható fájlt - például az árnyék jelszófájlt - megnyit, akkor abszolút útvonalnevet használhat, feltételezve, hogy a tényleges gyökér az abszolút gyökér.
- Előfordulhatnak olyan esetek, amikor a megbízható programnak utódjaira vonatkozóan korlátozottabb umask-ot kell alkalmaznia.
- Az utód folyamatok számos folyamatattribútumot örökölnek. Ha egy megbízható program tudatában van annak, hogy az utód folyamat megbízhatatlan, MAC címkéje a megbízható folyamat MAC címkéjével szemben nem dominál, illetve ezeket az attribútumokat a megbízható program megbízhatatlan őstől örökölte, akkor az attribútumok potenciálisan nem látható csatornák forrásai lehetnek.
- Ismerje a **fork** és **exec** rendszerhívások jogosultságterjesztésével kapcsolatos szabályoknak. A **fork** rendszerhívás alkalmával a szülőfolyamat jogosultságai az utódfolyamat jogosultságai lesznek. Az **exec** rendszerhívások során a jogosultságok megváltoznak.

A különösen érzékeny helyzetekben a megbízható program megvizsgálhatja a megbízható fájl hozzáférés-felügyeleti tulajdonságait annak biztosítására, hogy a fájl a megbízhatatlan programok által végrehajtott módosítások ellen kellően védett. A fájloknak például kötelezővé tehető, hogy a fájl tulajdonosa a root legyen, és a fájl tulajdonosa legfeljebb DAC írási jogosultsággal rendelkezzen.

Tényleges gyökér környezetek:

A megbízható programok a helyes abszolút útvonalnevekre támaszkodnak. A **login** program például arra támaszkodik, hogy az /etc/security/passwd fájl a helyes árnyék jelszófájl.

Ez nem csak az adatfájlokra igaz, hanem a megbízható programok végrehajtható fájljaira is. Ugyan a megbízhatatlan programok a **chroot** rendszerhívás segítségével a program tényleges gyökér könyvtárát közvetlenül nem tudják módosítani, létezhetnek olyan helyzetek, amikor a TCB lehetővé teszi a megbízhatatlan programoknak, hogy egy tényleges gyökér alatt fussanak. Ha ezek a megbízhatatlan program végrehajthatnak abszolút útvonalnevekre támaszkodó megbízható programokat, akkor potenciális biztonsági problémák állnak fenn.

Hitelesítés valódi és hatályos azonosítókkal:

A megbízható programoknak a folyamatokhoz tartozó számos felhasználói és csoportazonosító használatára szükségük lehet. Fontos tehát megismerni az azonosítók közti különbségeket, illetve az azonosítók megfelelő használatát.

Valós felhasználói és csoportazonosítók

A valós felhasználói és csoportazonosítók rendszerint annak a bejelentkezési munkamenetnek a bejelentkezési azonosságát képviselik, amelyben a folyamat létrehozásra került. Bizonyos esetekben a valós azonosítók (különösen a valós felhasználói azonosító) használhatók biztonsági döntések meghozatala során. Az egyik ilyen eset a felhatalmazásellenőrzés. A valós felhasználói azonosítókat a parancsok az azonosságellenőrzés egyik formájaként használják. Ez különösen hasznos a **setuid-on-exec** és **setgid-on-exec** vezérlőbitek rosszindulatú vagy felelőtlen használatának megakadályozásában. Azonban a valós azonosítók ellenőrzése eltér az általános UNIX gyakorlattól, tehát használatuk csak akkor javasolt, ha erre tényleg szükség van. A UNIX rendszereken használt átfogó elv szerint a hozzáférés ellenőrzése, illetve az egyéb kapcsolódó biztonsági ellenőrzések során a hatályos azonosítók használatosak. Az elfogadott gyakorlattól eltérni csak alapos megfontolás és részletes dokumentálás után tanácsos.

Tényleges felhasználói és csoportazonosítók

Az összes hozzáférés-felügyeleti döntésben (DAC és MAC) a hatályos felhasználói és csoportazonosítók használata javasolt. A rendszer felhasználói 0-127 közötti felhasználói azonosítóval rendelkeznek. A normál felhasználók azonosítóértéke 128, illetve ennél nagyobb.

Megbízható parancsok abszolút elérési útja:

Bizonyos biztonsági behatolási sémák hamis megbízható programokat próbálnak meg létrehozni, majd ezeket egy adminisztrátor - vagy akár normál - felhasználó által használt parancsértelmező-szerű program keresési útvonalán elhelyezik. A **passwd** parancs hamis változata például használható egy meglévő vagy új felhasználó jelszavának lementésére.

Az ilyen esetek megelőzéséhez a jó adminisztrációs gyakorlat az aktuális munkakönyvtár eltávolítása a keresési útvonalból. Azonban létezhetnek egyéb keresési útvonalak is, amelyek nem feltétlenül szigorúan védettek, illetve a normál felhasználók számára lehetővé kell tenni, hogy az aktuális munkakönyvtárat saját keresési útvonalukhoz adják. Hatékony ellenszer, hogy a megbízható programok mindig az abszolút útvonalnévvel kerüljenek meghívásra (például: /usr/bin/passwd). A megbízható program maga ellenőrzi első meghívási argumentumát, illetve a meghívási nevet. Ha nem a megfelelő abszolút elérési út kerül felhasználásra, akkor a megbízható program nem fog elindulni. A megbízható programnak ezen kívül biztosítania kell azt is, hogy ne rendelkezzen az abszolút gyökértől eltérő tényleges gyökérkönyvtárral.

Megjegyzés: A megoldás csak akkor hatásos, ha a felhasználókat kiképzik az abszolút útvonal kiadására. Ha egy felhasználó véletlenül relatív útvonalnevet használ, és hamis program kerül meghívásra, akkor a biztonsági behatolási séma kiküszöbölése sikertelen lesz.

Könyvtárfa strukturálása:

A könyvtárfák átgondolt strukturálásával kiterjeszthető a kritikus fájlok védelme. Az alapvető irányelv, hogy a könyvtár keresési hozzáférésének a lehető leginkább korlátozottnak kell lennie (például az összes nyilvánosan hozzáférhető fájlt a fájlrendszer-gyökérhez közeli könyvtárban ajánlott elhelyezni).

Jó ötlet - ezen kívül - a nagyon érzékeny könyvtárakat a lehető legközelebb helyezni az abszolút gyökérhez, mivel ez minimálisra csökkenti a védelmet igénylő köztes könyvtárak számát.

Csak olvasható fájlrendszerek:

Talán a könyvtárfa-strukturálásának legfelső foka, amikor a ritkán módosított megbízható fájlokat külön fájlrendszeren helyezik el, majd a fájlrendszert csak olvashatóként építik fel. Ezzel lényegében biztosítható, hogy a rendszer szokásos

üzemeltetése mellett a fájlok tartalma ne legyen módosítható. Az eljárást gyakran használják a megbízható programokhoz tartozó végrehajtható fájlok nagyméretű gyűjteményei esetében.

Ha egy fájlt módosítani kell, akkor a fájlrendszer írhatóként egy védettebb kontextusban (például egyfelhasználós módban vagy egy különálló, védettebb számítógépen) újra felépíthető. Ajánlott az ilyen frissítések után a fájlrendszer megfelelő beállítását (például a megfelelő DAC, MIC és MAC címkék meglétét) programok segítségével ellenőrizni.

Az előzőeken kívül a csak olvasható fájlrendszereken a DAC, MIC és MAC információk sem módosíthatók. Miután a fájlrendszerek megfelelően beállításra kerültek, az eljárás segítségével megvédhetők a különböző biztonsági behatolási sémáktól, amelyek megkísérlik a DAC információk és/vagy a MIC és MAC címkék módosítását.

Jelszókezelés:

Rendszerint nem minősül jó gyakorlatnak, hogy a szabványos rendszersegédprogramok a felhasználó bejelentkezési jelszavára rákérdezzenek. A jelszavak rendkívül érzékeny információknak minősülnek, tehát kezelésüket ajánlott szigorúan a néhány megbízható rendszersegédprogram körére korlátozni.

Bizonyos megbízható alrendszerek esetében szükség lehet saját jelszavakat megvalósítására. Azonban veszélyes lehet az ilyen saját jelszósémákra támaszkodni, hiszen ezek a rendszer által foganatosított mechanizmusokhoz képest kevésbé biztonságosak.

Megbízható számítástechnikai alapkörnyezet (TCB) védelme:

A TCB elemeit tartalmazó fájlokat védeni kell nem megbízható programok általi módosítás és néhány esetben felfedés (olvasás) ellen.

A módosítás elleni védelem kritikus, a felfedés elleni pedig kritikus lehet. A védendő fájlok a következők:

- Megbízható program által biztonsági döntés hozásához használt adatokat tartalmazó fájlok (például az árnyék jelszófájl)
- Megbízható program végrehajtható fájljai
- A TCB részeinek elérését lehetővé tevő pszeudofájlok (például: /dev/kmem).

Megjegyzés: A rendszerinicializálási fájlokat (rc fájlok) különösen védeni kell a TCB részeként

Módosítás elleni védelem:

A jogosulatlan módosítások elleni védelem elsősorban a DAC információk megfelelő értékre állításával érhető el. Az ilyen fájlok tulajdonosa rendszerint egy rendszerfelhasználói azonosító, és az írási hozzáférés csak a fájltulajdonos számára engedélyezett.

A MIC - rendeltetése szerint - a módosítás elleni védelmet az objektumok integritásának védelmén keresztül biztosítja. Ha egy fájlhoz magas MIC címkét rendel, akkor az alacsonyabb MIC címkével rendelkező folyamatok a fájlt nem módosíthatják, törölhetik, illetve nem nevezhetik át. Ez a fájlok nem kívánatos módosításának megelőzésére használható ideális módszer.

Bizonyos esetekben a MAC használható a jogosulatlan módosítások elleni védelem biztosítására. Azonban a MAC - rendeltetése szerint - csupán a felfedés (olvasás) ellen véd, és a módosítás elleni védelem biztosítására nem igazán megfelelő. Az alapszintű MAC házirend nem akadályozza meg, hogy az alanyok a magasabb címkével rendelkező objektumokat módosítsák. Ugyan a közvetlen fájlírások esetében nem engedélyezett, bizonyos megbízható alrendszerek ezt lehetővé tehetik. Továbbá sok megbízható fájlt (például végrehajtható programfájlt) alacsony MAC címkén kell tartani, hogy általánosan hozzáférhetők legyenek. Ennek következtében a fájlon a magas MAC címke beállítása nem mindig oldható meg.

Fájlok biztonsági kapcsolói szintén biztosítanak módosítás elleni védelmet. Bizonyos biztonsági kapcsolók még a privilegizált alanyok esetében is megakadályozzák az objektumok módosítását. Ha egy fájlra vonatkozóan az

FSF_TLIB fájlbiztonsági kapcsoló nincs beállítva, akkor a fájl csak akkor módosítható, amikor a rendszer konfigurációs módban van, feltéve, hogy a **trustedlib_enabled** kernelbiztonsági kapcsoló be van kapcsolva. Ahhoz, hogy egy fájlra vonatkozóan egy folyamat az **FSF_TLIB** kapcsolót beállíthassa, a folyamat EPS-ének tartalmaznia kell a **PV_TCB** jogosultságot. Egy másik érintett fájlbiztonsági kapcsoló az **FSF_APPEND** kapcsoló, amely megakadályozza a korábban írt adatok módosítását. Az olyan fájlokhoz, amelyek **FSF_APPEND** kapcsolója be van állítva, adatok csak hozzáfűzhetők. Ez hasznos lehet az olyan alkalmazások esetében, amelyek egy fájlban rekordokat rögzítenek.

A kapcsolókat a fájlok esetében többnyire az integrátorok állítják be és nem egy programvezérlő. A programozók számára ajánlott a kapcsolókkal, illetve funkciójukkal alapvetően tisztában lenni.

Felfedés elleni védelem:

A TCB fájlok az olvasási hozzáférés ellen a DAC és MAC segítségével védhetők. Az ilyen fájlokon beállított MAC címkék a fájlban tárolt információk érzékenységét tükrözik. Ha például egy bizonyos algoritmus minősített, akkor az algoritmust használó program végrehajtható fájlján beállított MAC címkét megfelelően be kell állítani.

Az adatok felfedése elleni védelme során elfogadható gyakorlat, hogy a MAC címke mesterségesen magasra (tehát a fájlban található adatok minősítésénél magasabbra) kerül beállításra. Azonban az ilyen megnövelt minősítéseket csak módjával szabad használni.

Szinte minden esetben - az abszolút gyökértől kezdődően - a teljes könyvtárláncot tanácsos védeni azért, hogy maga a fájl megfelelően védett legyen. Ellenkező esetben lehetséges, hogy egy rosszindulatú program képes lehet a könyvtárlánc egy részének leválasztására, illetve - a fájl hamis másolatának létrehozásával - új alkönyvtárfa létrehozására.

Tegyük fel például, hogy egy megbízható fájl az /A/B/foo helyen található. Ugyan a **foo** a módosítás ellen védett, a **B** könyvtár nem. Egy rosszindulatú, megbízhatatlan program képes lehet eltávolítani a **B** könyvtárban a **foo** fájlra mutató hivatkozást, és létrehozhat egy új **foo** fájlt a régi **foo** fájl hamis másolataként. A /A/B/foo fájlt megnyitó megbízható programokat ilyenkor észrevétlenül rávették a hamis adatok felhasználására.

A megbízható programok a TCB fájlok elérése során a helyes abszolút útvonalnevekre támaszkodnak. Ezen okból kifolyólag a TCB fájlok útvonalneveiben használt szimbolikus hivatkozásokat ajánlott ugyanolyan erősen védeni, mint magukat a fájlokat.

Bizonyos esetekben a MIC használható a jogosulatlan felfedések elleni védelem biztosítására. Azonban a MIC - rendeltetése szerint - elsősorban csak a módosítás (írás) elleni védelemre szolgál, és a felfedés elleni védelem biztosítására nem megfelelő.

Érzékenységi címkékkel kapcsolatos műveletek:

Az eltérő érzékenységi címkével rendelkező alanyokat és objektumokat érintő helyzetekre vonatkozóan léteznek megbízható programozási irányelvek.

Ajánlott ismerni az érzékenységi címkék formátumát, illetve a köztük fennálló dominanciaviszonyt. Ha a címke magasabb a másiknál, akkor az dominál, ha alacsonyabb, akkor a címkével szemben a másik címke dominál. A kiemelés azt jelenti, hogy az adatok osztályozását magasabb szintű címkére emeljük, a visszaléptetés pedig azt, hogy az adatok osztályozását alacsonyabb szintű címkére csökkentjük.

Alapszintű MAC megszorítás:

Az alapszintű kötelező hozzáférés-felügyeleti (MAC) megszorítás értelmében az 'A' érzékenységi címkén található adatokat a megbízhatatlan alanyok nem címkézhetik át a 'B' címkére, kivéve, ha a 'B' az 'A' címkével szemben dominál.

Az alapszintű MAC megszorítás mindent adatosztályra érvényes. A megszorítás korlátozza az adatok újracímkézését (tehát az adattároló címkéjének módosítását), illetve a címkével ellátott adatok mozgását az adattárolók között.

A rendszer különböző szintjein (rendszerhívás, rendszerszolgáltatási segédprogramok stb.) az alapszintű megszorítás átalakításra kerül konkrétabb szabályhalmazokra, azonban az alapötlet változatlan marad: az adatok legfeljebb kiemelésre kerülhetnek. A bővítés első szintje például, hogy a folyamatok olvasásra az objektumok nagy osztályának tetszőleges elemét akkor nyithatják meg, ha a folyamat címkéje az objektum címkéjével szemben dominál, illetve írásra akkor, ha az objektum címkéje a folyamat címkéjével szemben dominál.

Normál fájl esetében az írási műveletek a folyamattal egyező fájlokra korlátozottak. A könyvtárak és eszközök esetében az írási műveletek akkor engedélyezettek, ha az alany SL címkéje dominál az objektum minimális SL címkéjével szemben, illetve az objektum maximális SL címkéje dominál az alany SL címkéjével szemben. FIFO különleges fájlok (nevesített csővezetékek) esetében az olvasási műveletek korlátozottak - a nem látható csatornák miatt - a folyamattal egyező címkéhez tartozó FIFO különleges fájlokra.

Ugyan az adatok átállíthatók magasabb érzékenységi címkére, a képesség használata nem feltétlenül kötelező egy adott objektum és helyzet esetében. Maga az operációs rendszer például nem engedélyezi egy magasabb címkéjű fájl megnyitását írásra, bár az alapszintű MAC megszorítás szerint ez megengedett. Annak kérdése, hogy ez a megbízhatatlan alanyokra irányuló frissítés engedélyezett legyen-e, többnyire tervezési és elvi kérdés. Bizonyos esetekben hasznos lehet, míg egyéb helyzetekben nem. A magasabb címkével rendelkező fájlok közvetlen írásával kapcsolatosan felmerülő probléma például, hogy a folyamat ezeket a fájlokat nem olvashatja, tehát a magasabb szintű címkével rendelkező fájlok írása nem igazán hasznos. Azonban egy egyszerű megbízható segédprogram, amely egy megbízhatatlan alany kérésére kiemeli egy fájl címkéjét, elfogadható és hasznos segédprogram lehet.

A rendszerhívások szintjén a megszorítás csak a nem privilegizált folyamatokra vonatkozik, tehát a privilegizált folyamatokat a megszorítás nem köti. Azonban gyakorlatilag a megbízható rendszer által végrehajtott valamennyi szolgáltatást megbízhatatlan felhasználók számára tervezik, tehát a felhasználói szolgáltatások szintjén a megszorítás előtérbe kerül.

Az alapszintű MAC megszorítás a megbízhatatlan programoknak az adatátvitelre rendelkezésre álló összes módjára vonatkozik. Azonban az alapszintű MAC megszorítás gyakran két összetevőre oszlik. Az első összetevő csak az operációs rendszer adatátviteli (vagy címkekezelési) szolgáltatásaival foglalkozik. Az ilyen szolgáltatások - többek között - a fájlok írása és olvasása, illetve a folyamatok közötti adatkommunikáció. A második összetevő a nem tervezett kommunikációs módok, vagyis az ún. nem látható csatornák. A nem látható csatornák esetében az alapszintű MAC megszorítás tökéletes foganatosítása gyakorlatilag lehetetlen. Ebből az okból kifolyólag az alacsony adatsebességgel (például 0,1 bps) rendelkező nem látható csatornák létezhetnek, bár csak akkor, ha ezt más tényezők indokolttá teszik.

Az alapszintű MAC megszorítás magától érthetődő és egyszerű, és a többszintű adatok kezelésére kevés részletes irányelv létezik.

Többszintű műveletek:

A **sec_setplab** rendszerhívás segítségével egy privilegizált folyamat folyamatazonosítóját tetszőlegesen megváltoztathatja.

Mivel szinte az összes nem privilegizált folyamatra vonatkozó MAC és MIC megszorítás foganatosításra kerül a korábban létező rendszerhívásokra épülő privilegizált (tehát az alapszintű operációs rendszerben meghatározott) folyamatok esetében is, a többszintű műveletek végrehajtását igénylő privilegizált folyamatoknak a **sec_setplab** rendszerhívást kell használniuk. Azonban a megbízható programok a sec_setplab() rendszerhívást csak az alábbi módon használhatják:

- A **sec_setplab** rendszerhívást többszintű művelet végrehajtására (például magasabb címkével rendelkező fájlok megnyitása olvasásra) csak olyan függvénytár-rutinokon keresztül használhatók, amelyek tükrözik a végrehajtott tényleges, magas szintű művelet szemantikáját és elrejtik a **sec_setplab** rendszerhívás részletes használatát.
- Az egyetlen kivételt az olyan nagyon egyszerű folyamatcímke-módosítások alkotják, amelyek nem képezik nagyobb többszintű művelet részét. Ezek az egyszerű műveletek a **sec_setplab** rendszerhívást közvetlenül is használhatják.

A **sec_setplab** rendszerhívást érintő irányelveknek két oka van. Először is, a **sec_setplab** rendszerhíváshoz hasonló érzékeny és potenciálisan veszélyes szolgáltatásokat csak jól tervezett, moduláris módon használhatók. Másodszor, a

megbízható rendszerekre vonatkozó szabványok fejlődésével párhuzamosan lehetséges, hogy az alacsony szintű rendszerhívások a többszintű műveletek számos mechanizmusát támogatják.

A magas szintű műveletek függvénytárrutinokba foglalása kitűnő felfelé kompatibilitást és adaptálhatóságot biztosít az operációs rendszerek folyamatosan fejlődő változataival, illetve biztosítja a programok átírhatóságát a UNIX rendszer megbízható változatai között.

A megbízható rendszer biztosítja az ilyen rutinok alapvető halmazát. Amikor csak lehetséges, ezeket a rutinokat kell használni. A rutinhalmazt a későbbi operációs rendszer változatokkal párhuzamosan kell bővíteni. Amennyiben szükséges, ilyen függvénytárrutinokat a megbízható rendszer programozója is létrehozhat.

A MAC és MIC megszorítások másik kivétele az egy vagy több rendelkezésre álló MAC és MIC jogosultság használata a MAC vagy MIC megszorítások kihagyására. Az ilyen jogosultságok használatakor körültekintően kell eljárni.

System V Interprocess Communication (IPC):

Az IPC mechanizmusokra (üzenetsorok, szemaforok és osztott memória) DAC, MIC és MAC korlátozások érvényesek. Normális esetben nem állnak rendelkezésre parancsok System V objektumok létrehozásához és használatához.

Az AIX IPC-vel kapcsolatos rendszerhívások módosítva lettek a Trusted AIX rendszerhez, hogy több szintet tudjanak kezelni. Ezek a módosított rendszerhívások a következők:

- **msgget**
- **msgsnd**
- **msgrcv**
- **msgctl**
- **semget**
- **semop**
- **semctl**
- **shmget**
- **shmctl**
- **shmat**
- **shmdt**

Ezen felül a következő rendszerhívások kifejezetten a Trusted AIX rendszerhez adott IPC objektumok MAC attribútumainak kezeléséhez lettek kialakítva:

sec_getmsgsec
: Az üzenetsorok biztonsági attribútumainak lekérése

sec_getsemsec
: A szemaforok biztonsági attribútumainak lekérése

sec_getshmsec
: Az osztott memóriaszegmensek biztonsági attribútumainak lekérése

sec_setmsglab
: Az üzenetsorok biztonsági attribútumainak lekérése

sec_setsemlab
: A szemaforok biztonsági attribútumainak beállítása

sec_setshmlab
: Az osztott memóriaszegmensek biztonsági attribútumainak beállítása

A folyamatok számára az IPC objektumok kezeléséhez szükséges jogosultság-követelményekkel kapcsolatban tekintse meg az IPC objektumok elérése című részt. A **settxattr** parancs segítségével kezelhető az IPC attribútum.

Megvalósítási magas és rendszer magas MIC és MAC címkék:

Gyakran szükség lehet arra, hogy a megbízható folyamatok meghatározzanak egy olyan MAC címkét, amely a rendszer összes többi címkéjével szemben dominál. Két különböző MAC címke használható, a megvalósítási magas MAC címke, illetve a rendszer magas MAC címke.

A megvalósítási magas MAC címke a Trusted AIX által támogatott legmagasabb szintű MAC címke. A címke valószínűleg hierarchikus besorolással rendelkezik, illetve a telephelyen használatban nem lévő kategóriákat tartalmaz. A címke könnyen előállítható, azonban használata óvatosságot igényel. Nem ajánlott, hogy a folyamatok ezzel a címkével objektumokat hozzanak létre.

A rendszer magas MAC címke a telephely esetében használt legmagasabb MAC címke. A címkét az adminisztrátor a **LabelEncodings** fájlban határozza meg.

A rendszer magas MAC címke használata kevésbé hatékony, de használata mégis ajánlott, mivel az adminisztrátor hatékonyan korlátozhatja még a privilegizált folyamatok tevékenységeit is a **LabelEncodings** fájl kívánt paramétereinek megfelelő beállításával.

A MIC hasonlóképpen rendelkezik megvalósítási magas és rendszer magas címkékkel.

Felhasználó- és rendszerbejelentkezési tartományok:

Megbízható programok esetén, amelyek szolgáltatásokat hajtanak végre a felhasználók számára, szükség lehet a műveletek által érintett MIC és MAC címkék korlátozására olyan értékekre, amellyel a felhasználó bejelentkezhet és/vagy rendszerszintű engedélyezett bejelentkezési címkékre.

A rendszeren lévő felhasználókhoz rendelt engedélyek a **user** adatbázis /etc/security/user fájljában találhatók, és a **getuserattr** valamint **getuserattrs** függvénytár-rutinokkal érhetők el.

A Trusted AIX lehetővé teszi, hogy a felhasználók bármely olyan címkével működhessenek, amelyek a rendszerakkreditációs tartományban fel vannak sorolva és a minimális engedéllyel szemben dominálnak, valamint amelyekkel szemben a felhasználó maximális engedélye dominál. A programoknak, amelyek lehetővé teszik, hogy a felhasználók különböző címkékkel működjenek, mindig biztosítaniuk kell, hogy az új címke érvényes legyen a felhasználóhoz.

Tételezzünk fel például hogy egy **upgrade** nevű segédprogram lett megadva a fájlon lévő MAC címke növelése érdekében tetszőleges felhasználó általi kérésre. Az alapvető MAC korlátozási igény, hogy az **upgrade** csak azokat a fájlokat fogadja el, amelyek MAC címkéjével szemben a felhasználóé dominál. Továbbá biztonsági szempontból jó gyakorlatnak minősülhet (azonban nem szigorúan következik az alapvető MAC korlátozásból), hogy az új címkének olyannak kell lenniük, amelyekkel a felhasználó bejelentkezhet, és amelyek felhasználónkénti illetve rendszerszintű címketartomány-korlátozásokat foglalnak magukban. Az **upgrade** segédprogram az **sl_cmp** és **accredrange** felületet is használja erre a célra.

Könyvtárfaszerkezet:

A rendszer függvényeket hív meg, így a jogosulatlan folyamatok által létrehozott könyvtárfák nem csökkenő címkestruktúrát követnek, ahol a fájl címkéje megegyezik a szülőkönyvtáréval a particionált könyvtár tartományán belül, és a könyvtár címkéje dominál a szülőkönyvtáréval szemben (ne feledje el, hogy a dominancia az egyenlőséget is magában foglalja). Ez nem megbízható program természetes struktúrája.

A privilegizált folyamatokat ez a megszorítás nem köti és létrehozhatnak olyan könyvtárfákat, amelyben a szülőkönyvtár MAC címke kapcsolatai tetszőlegesek. Ilyen konfigurációk akkor hasznosak, ha a MAC keresés a fa gyökeréhez közelebb le van tiltva. Az összesítésvédelem például, ahol az adatobjektumok gyűjteményének MAC

címkéje nagyobb, mint az objektumok egyedülálló címkéi, úgy valósítható meg, ha a könyvtár MAC címkéjét az elemeinél nagyobbra állítja. A nem megbízható folyamatoknak domináns könyvtárcímkével kell rendelkezniük az adatok összesítésének eléréséhez.

A csökkenő címkével rendelkező könyvtárfák létrehozásánál körültekintően kell eljárni. Jogosulatlan folyamat nem nyithatja meg a fájlt írásra, ha a fájl nem domináns a szülő címkéjével szemben, vagy nem egyenlő vele.

Particionált könyvtárak kezelése:

a particionált könyvtárak megvalósításának eredményeként számos rendszerhívás viselkedése a megszokottól eltérő.

A particionált könyvtárak megvalósításának eredményeként az alábbi rendszerhívások viselkedése a megszokottól eltérő:

- getdirents
- link
- mkdir
- mount
- rename
- rmdir
- stat
- lstat
- fstat

Folyamatmód:

A **pdmode** parancs a megadott móddal hajtja végre a parancsot. A folyamat a **setppdmode** rendszerhívás segítségével beállíthatja a saját módját valós vagy virtuális módra. A **setppdmode** rendszerhívás sikerességéhez **PV_PROC_PDMODE** jogosultság szükséges. Egy folyamat nem módosíthatja más folyamat módját.

Könyvtártípus:

A **pdset** parancs segítségével egy normál könyvtár particionáltra módosítható, de a particionált könyvtár (vagy particionált alkönyvtár illetve al-alkönyvtár) nem módosítható normál könyvtárra.

A **pdmkdir** rendszerhívás segítségével is létrehozhatók particionált könyvtárak. A **pdmkdir** rendszerhívás **PV_FS_PDMODE** jogosultságot igényel.

MIC és MAC címketényezők:

Minden program csak az **sl_cmp** és **tl_cmp** függvényt használhatja a MIC és MAC címke közötti kapcsolat meghatározásához.

Ez különösen fontos, mivel a belső címkeformátum a későbbi rendszerváltozatokban módosulhat és ezek a függvénytár-rutinok követik a kialakuló formátumokat. Ehhez hasonlóan számos más függvénytár-rutin van, amely kezeli a MIC és MAC címkét. Ezek használni kell, amikor csak lehetséges.

A **setea**, **lsetea** és **fsetea** rendszerhívások módosítják a fájl MIC vagy MAC címkéjét. Az **fsetea** rendszerhívás fájlleírót fogad.

Illesztőprogramok:

A Trusted AIX rendszerekre íródott illesztőprogramok létrehozása során tanácsos néhány irányelvet figyelembe venni. Ismernie kell az alaprendszerhez szánt illesztőprogramok létrehozásának mechanizmusát, illetve a mechanizmusok használatával kapcsolatos óvintézkedéseket.

Eszközkezelő alrendszer:

Az AIX rendszereken található eszközök absztrakciók, amelyek az eszköz különleges fájlokra hivatkozva lefedik az összes elért adatobjektumot. Bizonyos esetekben ezek az adatobjektumok tényleges fizikai eszközöket képviselnek, ám bizonyos esetekben ettől meglehetősen eltérnek (például a /dev/null esetében, ahol adattároló objektum egyáltalán nem létezik). Az ilyen egyedeket gyakran pszeudó-eszközöknek hívjuk.

A Trusted AIX rendszerek két eszköztípust biztosítanak: az egycímkés, illetve a többszintes eszközöket. A többszintes eszközök egyidejűleg képesek megbízhatóan feldolgozni több, különböző érzékenységi szintű adatot. Az egycímkés eszközök rendszerint megbízhatatlanok. Az adatok címkéje rendszerint társításra kerül a többszintes eszközök által kezelt információkkal, olyan módon, hogy mindig biztosítva legyen az adatok megfelelő címkézése. Az egycímkés eszközök rendszerint külső címkekezelésre támaszkodnak.

A merevlemez például többszintes eszköz. A merevlemezen elhelyezett minden adathoz érzékenységi címke tartozik. A fizikailag olyan környezetben telepített nyomtató, amely a belépéshez biztonsági engedélyt igényel - például - egycímkés eszköz. A nyomtatóra csak az adott engedéllyel rendelkező adatok küldhetők.

Illesztőprogram-fejlesztési szempontok:

Az illesztőprogramok az operációs rendszer kernel részét képezik, és ezáltal tevékenységük nem korlátozott. Az illesztőprogramok létrehozása és módosítása pont annyira érzékeny terület, mint a kernel módosítása. Sajnálatos módon a felhasználóknak gyakran szükségük van az illesztőprogramok létrehozására vagy módosítására. A folyamat során különösen körültekintően kell eljárni.

Lehetetlen az összes konkrét szempontot felsorolni, amelyet szem előtt kell tartani az illesztőprogramok írásakor, mivel az illesztőprogramok számos - néha teljesen ártatlan - módon alááshatják a rendszer biztonságát. Ennek következtében a biztonságos illesztőprogramok létrehozása többnyire a tervezők ítélőképességén és tapasztalatán múlik.

Az illesztőprogramok az egyszerű eszközkezelésnél többet ne végezzenek. Az olyan illesztőprogramokat, amelyek elsődleges célja a rendszer új rendszerhívásokkal való bővítése (beleértve ebbe számos pszeudó-illesztőprogramot, például a /dev/kmem pszeudó-illesztőprogramjait), tanácsos új rendszerhívásoknak tekinteni, és ennek megfelelően tervezni. A szakaszban leírt irányelvek elsősorban azokra az illesztőprogramokra vonatkoznak, amelyek szabályszerű eszközkezelők.

Mielőtt új illesztőprogramokat próbálna meg létrehozni, tanácsos először a szabványos illesztőprogramokat tanulmányozni. Az illesztőprogramok elsődleges biztonsági tevékenységei az **open** és **ioctl** rendszerhívások végrehajtásában érintett tevékenységek.

Eszközök megnyitása:

Akárcsak a legtöbb rendszerobjektum esetében, az eszközhozzáféréssel kapcsolatos legtöbb biztonsági ellenőrzés akkor kerül végrehajtásra, amikor az eszközt az **open** rendszerhívással megnyitja.

A kernel először számos alapműveletet hajt végre, majd a megnyitási kérés feldolgozását átadja az illesztőprogramnak. Mielőtt a kernel a vezérlést az illesztőprogramnak átadná, az alábbi biztonsági ellenőrzéseket hajtja végre:

- Ha a folyamat az eszköz különleges fájlhoz nem rendelkezik MAC hozzáféréssel, akkor a megnyitás meghiúsul
- Ha a folyamat az eszköz különleges fájlhoz nem rendelkezik MIC hozzáféréssel, akkor a megnyitás meghiúsul
- Ha a folyamat az eszköz különleges fájlhoz nem rendelkezik DAC hozzáféréssel, akkor a megnyitás meghiúsul

Számos eszköz esetében az eszköz olvasása (a **read** rendszerhívás segítségével) az eszköz állapotát olyan módon változtatja meg, hogy az az olvasási folyamat MAC címkéje által nem dominált MAC címkével rendelkező folyamat által észlelhető. Ez potenciálisan nem látható csatornának minősül. A probléma a FIFO természetű eszközök esetében jelentkezhet. Az ilyen esetekben a bevett gyakorlat szerint az olvasási hozzáférést korlátozni kell az olyan folyamatokra, amelyek az eszközzel egyező MAC címkével rendelkeznek. Ez az illesztőprogramon belüli ellenőrzés segítségével végezhető el.

A szabályostól eltérő eszközök tervezésével kapcsolatosan kevés konkrét irányelv létezik. Ilyen esetekben fontos megérteni és alkalmazni a kötelező és tetszés szerinti hozzáférés-felügyelet alapelveit. Szerencsére a legtöbb illesztőprogram szabályos eszközként konfigurálható, a szabályostól eltérő illesztőprogramokkal járó különleges problémákkal a fejlesztő ritkán szembesül.

Illesztőprogram megnyitása példák:

Az alábbi szabványos rendszer-illesztőprogramokból származó példák bemutatják a nem szabványos eszközkezelést. A példák célja annak bemutatása, hogy az ilyen illesztőprogramoknak igen széles skálája létezhet.

/dev/null

A /dev/null adattárolóval nem rendelkező pszeudó-eszköz. A /dev/null helyre írt adatok eldobásra kerülnek, illetve az olvasási kérésekre válaszként mindig fájl vége (EOF) kerül visszaadásra. Ennek megfelelően megnyitásához nincs szükség MAC eszközmegszorításra. Kompatibilitási okokból a /dev/null eszközfájlon DAC hozzáférés szükséges, bár szigorúan ez sem szükséges.

/dev/tty

Amikor a /dev/tty eszközön open utasítást ad ki, akkor az illesztőprogram tulajdonképpen a kérő folyamat vezérlőterminálját próbálja megnyitni. Ennek következtében a MIC, MAC, illetve DAC hozzáférést a folyamat vezérlőtermináljára vonatkozóan, nem pedig a /dev/tty eszközre vonatkozóan kell ellenőrizni. Kompatibilitási okokból a /dev/tty DAC hozzáférést igényel, bár szigorúan ez sem szükséges.

ioctl korlátozások:

Ugyan minden illesztőprogram-felület funkciónak megbízhatónak kell lennie, az **ioctl** felület általában különleges figyelmet igényel.

Általános szabályként csak az írási hozzáféréssel rendelkező folyamatok módosíthatják egy fájl jellemzőjét úgy, hogy azt az egyéb, írási hozzáféréssel nem rendelkező folyamatok észlelhessék. Az írási hozzáférés azt jelenti, hogy a folyamat a fájlt írásra megnyitotta vagy a folyamat MAC címkéje és az eszköz címkéje egyenlő. Ez a korlátozás abból az alapszintű MAC megszorításból ered, hogy a folyamatok nem hajthatnak végre olyan tevékenységet, amelyet az alacsonyabb MAC címkén található folyamatok észlelhetnek.

Ha a tevékenység célja felhasználói olvasási/írási művelet, akkor a korlátozást a fenti módon foganatosítani kell. Ellenkező esetben az ilyen helyzeteket - amikor a korlátozás nem kerül foganatosításra - a rendszer nem látható csatornaként kezeli, tehát ezek sávszélességét korlátozni kell és/vagy a csatornáknak megfigyelhetőknek kell lenniük.

Bizonyos eszközfelügyeleti tevékenységeket szükséges lehet a privilegizált folyamatokra korlátozni akkor is, ha az eszköz nem megbízható.

Egyéb korlátozások:

Viszonylag kevés egyéb olyan helyzet létezik, amikor az illesztőprogramnak különleges biztonsági ellenőrzéseket kellene foganatosítania.

Az egyik ilyen példa, amikor egy eszköz olvasása az eszköz állapotát olyan módon változtatja meg, hogy az az olvasási folyamat MAC címkéje által nem dominált MAC címkével rendelkező folyamat által észlelhető. Ez potenciális nem látható csatornát jelölhet, amelyet magának az illesztőprogramnak kell korlátozni vagy megfigyelni.

Illesztőprogram-fejlesztési programozási összefoglalás:

Az illesztőprogramok megvalósításakor az alábbi irányelveket tanácsos szem előtt tartani.

Megjegyzés: Az adatfolyam- és FIFO eszközökön végzett írási/olvasási műveletek kiterjesztett biztonsági szolgáltatásainak támogatásához a rendszer új rendszerhívásokkal bővült. A kiterjesztett biztonság attribútumokat két új függvénytár API, az eread() és az ewrite() támogatja. Ha az attribútum MLS kernelben található, akkor az eszközön beállításra kerül a DEV_SEC_ERDWR biztonsági kapcsoló. FIFO esetében - hasonlóképpen - az eszközön beállításra kerül a GNF_SEC_ERDWR. A kapcsolók további biztonsági ellenőrzéseket tesznek lehetővé az egyes írási/olvasási műveletek során.

Általános tervezési eljárások

Az illesztőprogram minden biztonsági ellenőrzését moduláris módon ajánlott megírni, illetve úgy, hogy könnyen azonosíthatók legyenek.

Illesztőprogramokon belüli ellenőrzések

Ajánlott a MIC, MAC és DAC ellenőrzéseket az illesztőprogramon kívül tartani. Az ilyen ellenőrzések nélküli illesztőprogramok könnyen átírhatók megbízhatatlan rendszerekről és egyéb típusú megbízható rendszerekről, illetve ilyen rendszerekre.

A szabályos illesztőprogram-megvalósítások esetében a MIC, MAC és DAC ellenőrzéseket a kernel hajtja végre, az egyéb szükséges jogosultságellenőrzéseket maga az illesztőprogram végzi. A szabályostól eltérő illesztőprogram-megvalósítások esetében az összes ellenőrzés (MIC, MAC, DAC, illetve jogosultságellenőrzések) az illesztőprogramban kerül végrehajtásra. A szabályos vagy a szabályostól eltérő illesztőprogram-megvalósítás közötti választás nagyrészt tervezési döntés kérdése.

DAC

A DAC foganatosításra kerül minden eszköz speciális fájl esetében, az eszközhozzáféréshez használt fájlrendszer belépési pont alapján.

Megfelelő telepítés ellenőrzése

A MAC ellenőrzéseket végrehajtó minden illesztőprogramnak tanácsos biztonságosan kezelnie (az ésszerűség határain belül) azt a lehetőséget, hogy az eszköz meghatározása nem megfelelő.

Privilegizált hozzáférés

Lehetséges, hogy egy illesztőprogram esetében nem alkalmazható, hogy az eszközműveleteket a privilegizált folyamatokra korlátozza. Azonban ezekre a helyzetekre is létezik néhány konkrét tanács.

Annak meghatározására, hogy rendelkezik-e a szükséges jogosultságokkal, a **refmon** kernelfunkció használható.

Legkevesebb jogosultság:

A Trusted AIX bevezeti a legkevesebb jogosultság fogalmát. A legkevesebb jogosultság a korábban széleskörű jogosultságokkal rendelkező root felhasználót kifinomultabb jogosultsági mechanizmusokra osztja. A jogosultságok ilyen formájú felosztása biztosítja, hogy a megbízható szoftverben előforduló programozási vagy egyéb hiba esetén a rendszerbiztonság csak minimális mértékben sérülhet.

Jogosultságműveletek:

Az egyes folyamatokhoz négy jogosultságvektor tartozik: hatályos, maximális, örökölhető, illetve korlátozó.

A maximális jogosultságvektor meghatározza az egyes folyamatok aktív jogosultságainak felső határát. A hatályos jogosultságvektor meghatározza a jogosultsággal kapcsolatos döntések meghozatala során használt jogosultságokat. Fontos megjegyezni, hogy a hatályos jogosultsághalmaz mindig a maximális jogosultsághalmaz részhalmaza, amely viszont mindig a korlátozó jogosultsághalmaz részhalmaza. A korlátozó jogosultsághalmaz meghatározza azokat a

jogosultságokat, amelyekkel a folyamat maximális, örökölhető, illetve hatályos jogosultsághalmazai rendelkezhetnek. Az örökölhető jogosultsághalmaz az utódfolyamatok által a fork és exec rendszerhívások között örökölhető jogosultsághalmaz.

Új szövegkép végrehajtásakor a jogosultságok kiterjesztése a következő algoritmus alapján történik. Az említett különleges jogosultságok a **PV_ROOT**, a **PV_SU_**, a **PV_SU_EMUL**, a **PV_SU_ROOT**, a **PV_AZ_ROOT** és a **PV_SU_UID**.

Az alábbi algoritmus a legkevesebb jogosultság alrendszer két fontos alapelvét mutatja be. Az első alapelv szerint csak a különleges jogosultságok (**PV_ROOT**, **PV_SU_**, **PV_SU_EMUL**, **PV_SU_ROOT**, **PV_AZ_ROOT** és **PV_SU_UID**) terjeszthetők feltétel nélkül egy új folyamatképfájl végrehajtásán keresztül. A második alapelv szerint a folyamat hatályos jogosultságvektora kiürítésre kerül, hacsak a fájl **FSF_EPS** biztonsági kapcsolója nincs beállítva. Ez biztosítja a visszamenőleges kompatibilitást az olyan alkalmazásokkal, amelyeket a megbízható rendszeren futtatnia kell anélkül, hogy fel lennének készítve a legkevesebb jogosultság rendszerre.

```
new_max_privs = old_inheritable_privs
new_max_privs = new_max_privs | file_innate_privs
IF (a fájl PAS halmazban a felhasználóhoz felhatalmazások kerültek
hozzárendelésre)
new_max_privs = new_max_privs | file_authorized_privs
new_max_privs = new_max_privs & old_limiting_privs
IF (old_max_privs legalább egy különleges jogosultságot tartalmaz)
new_max_privs += ugyanaz a különleges jogosultsághalmaz
IF (a végrehajtható fájlra beállításra kerül az FSF_EPS)
new_eff_privs = new_max_privs
ELSE
new_eff_privs = old_inheritable_privs
IF (old_eff_privs legalább egy különleges jogosultságot tartalmaz)
new_eff_privs += ugyanaz a különleges jogosultsághalmaz
new_limiting_privs = old_limiting_privs
```

Jogosultságok adása és eltávolítása:

Az alábbi szabványos függvénytárrutinok bemutatják a rendszer jogosultságainak kezelését. A rutinok csak a rendszer privilegizált programjaihoz használhatók.

priv_raise
: A folyamat hatályos jogosultságvektorának módosítása a megadott jogosultságlista felvételével (vagy növelésével). A jogosultságlistát a folyamat maximális jogosultságvektorának kell tartalmaznia, ellenkező esetben a rendszer hibát jelez.

priv_remove
: A folyamat hatályos és maximális jogosultságvektorának módosítása a megadott jogosultságlista eltávolításával. Ha a folyamat a hatályos vagy maximális jogosultságokat nem tudja eltávolítani, akkor a rendszer hibát jelez.

priv_lower
: A folyamat hatályos jogosultságvektorának módosítása a megadott jogosultságlista eltávolításával (vagy csökkentésével). Ha a folyamat a hatályos jogosultságokat nem tudja csökkenteni, akkor a rendszer hibát jelez.

A rutinok - bemenetként - jogosultságok vesszővel elválasztott listáját fogadják el, amelyet **-1** (mínusz egy, érvénytelen jogosultságszám) zár le. A jogosultságokat igénylő legkisebb kódrészlet jogosultságait növelő vagy csökkentő módszert jogosultságbefogásnak hívjuk. A rossz tervezésű vagy kivitelezésű szoftverek okozta biztonsági sértések előfordulásának csökkentéséhez az összes megbízható alkalmazásnak tanácsos jogosultságbefogást alkalmaznia.

setppriv
: A folyamat hatályos, maximális, örökölhető, illetve korlátozó jogosultságvektorának módosítása a jogosultsághalmazok beállításával. Ha az átadott jogosultsághalmazok érvénytelenek vagy nem engedélyezettek, akkor a rendszer hibajelzést ad vissza.

Felhatalmazások:

A felhatalmazások a bizonyos felhatalmazással rendelkező felhasználók számára különböző jogosultsághalmazokat biztosítanak.

Általában a parancsok és segédprogramok az érintett felhatalmazásokat a végrehajtás kezdetekor ellenőrzik, majd saját jogosultságaikat ennek tükrében állítják be. Ennek következtében az adott jogosultsággal rendelkező felhasználók - a parancs programozása szerint - az egyes végrehajtott parancsokra vonatkozóan eltérő jogosultsághalmazt kapnak.

Ahhoz, hogy a meglehetősen fáradalmas jogosultságbeállítás magából a kódból eltávolítható legyen, az AIX biztosítja a bináris fájlhoz képest külső felhatalmazás- és jogosultsághalmazok használatát. A jogosult felhatalmazáshalmaznak (PAS) és a felhatalmazott jogosultsághalmaznak (APS) köszönhetően a rendszer - és nem maga a parancs - hajtja végre a jogosultságok beállítását a felhatalmazás alapján.

checkauths
Összehasonlítja az átadott felhatalmazáslistát és az aktuális folyamathoz tartozó felhatalmazásokat.

A felhatalmazás ellenőrzésével kapcsolatosan további információkat a “RBAC felhatalmazások” oldalszám: 84 témakör tartalmaz.

Megfigyelés:

A Trusted AIX nyomkövetési napló előállítására és a vonatkozó információk kezelésére számos parancsot biztosít. Nem valószínű, hogy a megbízható rendszer programozójának ezeket a programokat módosítania kellene, illetve új programot kellene felvennie.

audit A megfigyelési démon vezérlése

auditbin
A nyomkövetési naplófájlok vezérlése

auditselect
A nyomkövetési naplófájlokból származó megfigyelési rekordok összefésülése és kijelölése

auditpr
A kijelölt események megjelenítése felhasználó által olvasható formátumban

A megbízható programok által előállított megfigyelési események jelentik azt a legfontosabb területet, ahol a megfigyelés a megbízható rendszerek programozója számára fontos lehet. A legtöbb megbízható programnak üzeneteket kell kiadnia a rendszer nyomkövetési naplója felé.

Megfigyelendő helyzetek:

Kevés pontos irányelv létezik arra vonatkozóan, hogy a megbízható programok milyen helyzeteket észleljenek, illetve figyeljenek meg. A döntés elsősorban a józan ész és a megfigyelési stratégia függvénye. Az alaprendszer a helyzeteket sikerekre, meghibásodásokra, objektum-hozzáférésekre, illetve potenciális nem látható csatornákra osztja.

Sikerek:

Az alapvető használati előzmények kialakításához fontos a sikeres műveleteket is megfigyelni.

Fontos például, hogy egy eszközfoglaló program rögzítse, hogy egy adott felhasználó mikor foglal le, illetve szabadít fel egy eszközt. Az információk segítségével a program képes a rendszer információfolyamának nyomon követésére, és ezáltal a felelősség meghatározására akkor, ha a későbbiek során kiderül, hogy az eszközzel visszaélés történt. Másfelől bizonyos megfigyelési filozófiák nem szentelnek figyelmet a sikeres műveleteknek, mivel azok a megbízható szoftver szerint legálisak és megfelelők.

Meghibásodások:

A hibás műveletek megfigyelése hasznos lehet az olyan felhasználók kiszűrésére, amelyek nem engedélyezett szolgáltatásokhoz vagy adatokhoz kísérelnek meg hozzáférni. Az ilyen meghibásodások gyakori előfordulása rosszindulatú (ha nem is különösebben okos) személyekre utalhat.

Az alaprendszer a meghibásodásokat öt kategóriába sorolja:

- Jogosultsági hibák (nem privilegizált folyamat által egy privilegizált folyamatokra korlátozott tevékenység végrehajtására tett kísérlet)
- MAC hibák (a tevékenység meghiúsul, mert a MAC megszorításokat sértené)
- MIC hibák (a tevékenység meghiúsul, mert a MIC megszorításokat sértené)
- DAC hibák (a tevékenység meghiúsul, mert a DAC megszorításokat sértené)
- Egyéb hibák (például bejelentkezési kísérlet helytelen jelszóval)

Objektum-hozzáférések:

Az objektum-hozzáférés megfigyelése azért szükséges, mert ezáltal figyelhetők meg az adott objektumhoz (például az árnyék jelszófájlhoz) hozzáférő felhasználók.

Potenciális nem látható csatornák:

A potenciális nem látható csatornák megfigyelése azért fontos, mert a nem látható csatornák segítségével különböző MAC címkéken futó folyamatok között információk adhatók át. A potenciális nem látható csatornák használata nem jelenti azt, hogy a csatornákat erre a célra ténylegesen használták, csupán azt, hogy az ilyen felhasználás lehetséges.

A megfigyelési rendszer által létrehozott minden bejegyzés tartalmazza a megfigyelési bejegyzés okát (siker, MAC hiba, MIC hiba, DAC hiba, jogosultság hiba, egyéb hiba, objektum-hozzáférés, illetve potenciális nem látható csatorna). Ebbe beletartoznak a rendszer által létrehozott, illetve a felhasználói programok által létrehozott megfigyelési rekordok is.

Hasznos lehet átgondolni, hogy a felhasználó megbízható-e (tehát adminisztrátor-e), de nem létezik tökéletes módszer annak meghatározására, hogy a megbízható vagy a megbízhatatlan felhasználók igényelnek-e komolyabb megfigyelést. Az adminisztrátorokat - például - ugyan megbízhatónak feltételezzük és ilyen szempontból kevesebb megfigyelést igényelnének, tevékenységük azonban sokkal komolyabb következményekkel járhat, tehát hasznos lehet a jogosulatlan adminisztrátorok tevékenységeinek rögzítése. A normál felhasználók kevesebb kárt okozhatnak és ebben az értelemben kevesebb megfigyelést igényelnek, azonban ugyanakkor kevésbé megbízhatók, tehát több megfigyelést igényelhetnek. A rendszeradminisztrátorok saját tevékenységükre általában nagyobb fokú megfigyelést alkalmaznak, hogy biztonsági visszaélések esetén ártatlanságukat bizonyíthassák.

A következő eseményeket tanácsos megfigyelhetővé tenni:

- A sikeres - különösen az információátvitelt, illetve a hozzáférés-vezérlési paraméterek módosítását magukban foglaló - műveleteket
- A biztonsági okból meghiúsuló műveleteket
- Az adminisztrátori műveleteket, tekintet nélkül arra, hogy sikeresek voltak-e vagy sem
- A nem látható csatornák potenciális használatát
- Az adott objektumhoz hozzáférő műveleteket
- A tényleges nyomkövetési napló további tartalmát érintő tevékenységeket

Megfigyelési információs szintek:

A magas szintű megfigyelési információk hasznosabbak az alacsony szintű megfigyelési információknál. A megbízható programok a műveletek magas szintű nézetét biztosítják, és kiváló megfigyelési üzeneteket állítanak elő.

Csupán annak rögzítése, hogy egy adminisztrátor írásra megnyitott egy biztonsági fájlt, sokkal kevésbé hasznos, mint a fájlon végrehajtott magas szintű művelet rögzítése (például annak rögzítése, hogy egy adminisztrátor a fájlban új bejegyzést hozott létre, illetve az új bejegyzés kulcsinformációinak rögzítése). Tanácsos a megfigyelési információkat a lehető legmagasabb szintre beállítani.

Előnyösebb egy adott esemény információit rögzíteni, mint a több eseményre vonatkozó információkat. A megfigyelési előfordulás több esemény közötti felosztásának elsődleges célja, hogy a különálló előfordulások egymástól függetlenül engedélyezhetők legyenek.

Megfigyelési osztályok és események:

Minden megbízható programnak meg kell határoznia a megfigyelési osztályt, megfigyelési eseménytípust és okot, amelyet akkor használ, amikor a megfigyelési üzeneteket az **auditlog** rendszerhívás segítségével kiadja.

Minden megfigyelési esemény megfigyelési osztályhoz tartozik. Azáltal, hogy az eseményeket osztályokhoz rendeli, a nagyszámú esemény könnyebben kezelhető. A megfigyelési osztályok meghatározását az /etc/security/audit/config fájl tartalmazza.

Az események rögzítése a megfigyelési osztály segítségével engedélyezhető, illetve tiltható le. Ha fontos, hogy két eseményt egymástól függetlenül engedélyezzen, akkor ezeket az eseményeket nem szabad ugyanahhoz a megfigyelési osztályhoz rendelni. Általánosságban azonban az események osztályhoz rendelése jó gyakorlatnak minősül. Az esetek többségében minden megbízható program, illetve a hozzá tartozó megbízható programok egy - vagy kivételes esetben néhány - megfigyelésiosztály-nevet saját használatra tartanak fenn.

A megfigyelhető rendszertevékenységek az /etc/security/audit/events fájlban kerülnek meghatározásra, megfigyelési eseményként.

Nem látható csatornák:

A rendszer feltételezi, hogy a megbízható szoftverek nem látható csatornákat használó sémákban nem vesznek részt. A szoftvereket olyan módon kell megtervezni, hogy a megbízhatatlan szoftverek a nem megbízható csatornákat ne használhassák ki. Az alábbi szakasz tartalmazza a nem látható csatornák meghatározását, illetve az ilyen csatornák felismerésére és korlátozására vonatkozó irányelveket.

Nem látható csatornák meghatározása:

Az 'A' címkén futó folyamatok közül egyik sem hajthat végre a 'B' címkén futó folyamat által észlelhető tevékenységet, kivéve akkor, ha a 'B' címke az 'A' címkével szemben dominál.

A definíció két helyzetre bontható: a közvetlen adatműveletre, illetve a járulékos műveletekre. A közvetlen adatműveletek célja, hogy a felhasználók számára a felhasználói adatok tárolására és kommunikálására (például fájlírás és -olvasás) közvetlen módot biztosítsanak. A műveleteknek a legteljesebb mértékben meg kell felelniük az alapszintű MAC megszorításnak. Az összes többi művelet járulékos műveletnek minősül. Az ún. nem látható csatorna a járulékos műveletek olyan felhasználása, amely során a műveletek az alapszintű MAC megszorítás ellenében adatokat adnak át.

A nem látható csatorna kihasználása két megbízhatatlan folyamatot igényel, amelyekre küldőként (az 'X' címke helyen), illetve fogadóként (az 'Y' címke helyen) hivatkozunk. Feltételezzük, hogy a fogadó MAC címkéje nem dominál a küldő MAC címkéjével szemben (ellenkező esetben a küldő-fogadó adatfolyam legális kiemelés lenne). A csatorna kihasználásához mind a küldő, mind a fogadó felhasznál bizonyos - a megállapodás tárgyát képező erőforrások felhasználására vonatkozó - egyezményeket, hogy az adatokat a MAC megszorítás ellenében továbbítsa.

A nem látható kihasználás egyetlen feltétele, hogy a fogadó címkéje ne dominál̈jon a küldő címkéjével szemben, illetve hogy a küldő és fogadó egyaránt megbízhatatlan legyen. Gyakran mind a küldő, mind a fogadó ugyanannak a felhasználónak a nevében kerül felhasználásra. Feltételezzük, hogy maga a TCB figyelembe veszi az alapszintű MAC megszorítást, és nem tartalmaz a megszorítást a nem látható csatornák nem megfelelő felhasználása által sértő kódot. (Valójában a privilegizált folyamatoknak sokkal hatékonyabb módszerek állnak rendelkezésükre annál, hogy nem

látható csatornákra kelljen hagyatkozniuk.) Ami aggodalomra adhat okot, hogy a megbízhatatlan folyamatok képesek lehetnek a nem látható csatornákat megbízható programok segítségével kihasználni.

Általánosságban tanácsos a nem látható csatornákat eleve kizárni a rendszerből. Azonban előfordulhatnak olyan esetek, amikor az egyéb rendszerigények (például teljesítmény, megbízhatóság vagy kompatibilitás) túlságosan korlátozottak lennének a nem látható csatornák jelenléte nélkül.

Sávszélességre vonatkozó irányelvek:

Az alaprendszer a nem látható csatornák korlátozásához - a sávszélesség alapján - az alábbi irányelveket használja:

Több mint 100 bps
Ilyen csatornák nem létezhetnek

0,1-100 bps
A tartományba eső csatornák - ha mindenképpen szükségesek - létezhetnek, de használatukat a rendszer érzékeli, illetve - amikor lehetséges - megfigyeli

Kevesebb mint 0,1 bps
A tartományba eső csatornák - szükség esetén - létezhetnek, de használatukat különösebben nem szükséges érzékelni

Tanácsos az összes további TCB program esetében is ezeket az irányelveket követni. Ezen kívül érdemes megfontolni, hogy még a 10 bps sebességű, viszonylagosan lassú csatornák is óránként 4 500 byte átvitelére képesek, amely - ha figyelembe vesszük, hogy illegálisan kerül letöltésre - jelentős mennyiség. Ennek következtében mindent el kell követni azért, hogy a nem látható csatornák sávszélessége a lehető legkisebb legyen.

A legtöbb nem látható csatorna esetében a sávszélességet általában a csatornát kihasználó folyamatoktól eltérő folyamatok tevékenységei csökkentik. Azonban nem tanácsos arra építeni, hogy ezek a folyamatok korlátozzák a nem látható csatornák sávszélességét, hiszen minden rendszeren léteznek alacsony aktivitású időszakok.

Nem látható csatornák felismerése:

A nem látható csatornák felismerése többnyire az óvatos elemzés és tervezés kérdése. A nem látható csatornák felismerésére számos konkrét irányelv létezik.

A "modul" kifejezés a TCB kód egy olyan egységére utal, amely felismeri vagy korlátozza a nem látható csatornák használatát, a kernelben és a folyamatokban egyaránt. A nem látható csatornák felismerése elsősorban annak meghatározásából áll, hogy az 'A' szinten futó megbízhatatlan folyamat (a küldő) a 'B' szinten futó másik folyamat (a fogadó) által észlelhető tevékenységet hajt végre úgy, hogy a 'B' szint az 'A' szinttel szemben nem dominál.

Egy általános nem látható csatorna például a megbízhatatlan felhasználó nevében a megbízható folyamat által a fájlba adatok írása, amikor a fájl a felhasználó MAC címkéjét nem dominálja.

A nem látható csatornák felismerésére meglehetősen kevés módszer született. A legelterjedtebb az osztott erőforrásmátrix (SRM). Az eljárás leírását az alábbi kiadványok tartalmazzák:

- • Kemmerer, R.A. "Shared Resource Matrix Methodology: An Approach to Identifying Storage and Timing Channels," ACM Transactions on Computing Systems 1(3) 1983, 256-277.
- • Tsai, CR. "A Formal Method for the Identification of Covert Storage Channels in Source Code," Proceedings of the 1987 IEEE Symposium on Security and Privacy, 74-87.

Nem látható csatornák felismerése megfigyeléssel:

A fenyegetés hatékony ellenszere a nem látható csatornák potenciális használatának megfigyelése. A megfigyelés hasznosságához azonban a megfigyelési eseménynek meglehetősen ritkának kell lennie. A megfigyelés kevéssé hasznos akkor, ha a tényleges kihasználás és a megfigyelést okozó esemény járulékos használatának aránya alacsony.

Nem látható csatornák korlátozása:

A nem látható csatornák legjobb korlátozási módja maguknak a csatornáknak az eltávolítása.

Ellenkező esetben a csatornákat a Sávszélességre vonatkozó irányelvek szakaszban leírt irányelveknek megfelelően kell korlátozni. Ezen kívül - amennyiben lehetséges és hatásos - a csatornák potenciális használatát kell megfigyelni.

Általánosságban nehéz a kernel-, illetve illesztőprogramkódnak a nem látható csatornákat elkülöníteni és korlátozni, mivel mind a kernel-, mind pedig az illesztőprogramkód tervezése során a hatékonyság az elsődleges szempont, illetve csatornáik sávszélessége nagyobb. A megbízható folyamatok a nem látható csatornákat könnyebben korlátozhatják.

Megjegyzés: Az azonos címkéhez tartozó folyamatok, illetve az olyan folyamatok esetében, ahol a fogadó a küldővel szemben dominál, a nem látható csatornák felhasználására vonatkozóan korlátokat nem szükséges felállítani. Ennek köszönhetően a legtöbb TCB modul képes a rendszerteljesítmény javítására azáltal, hogy az ilyen esetekben nem foganatosítanak korlátozásokat.

Címkealapú kvóták:

Számos nem látható csatorna magában foglalja a különböző MAC címkéken futó folyamatok között megosztott erőforrástár használatát. Ezek hatékonyan korlátozhatók akkor, ha az egyes MAC címkékhez különálló, rögzített méretű erőforrástárakat hoz létre, hogy ezáltal a folyamatok csak a saját MAC címkéjükhöz tartozó tár erőforrás-használatát modulálhassák.

Az idő múlásával a fel nem használt erőforrások egyik tárból áthelyezhetők a másikba, hogy ezáltal kielégítsék a dinamikusan változó igényeket. Ez az erőforrás-átállítás maga is egy nem látható csatorna, de sávszélessége sokkal alacsonyabb és könnyen korlátozható.

Időkésleltetések:

A nem látható csatornák korlátozásának egyik módszere szerint a TCB biztosítja, hogy egy bizonyos mennyiségű idő elteljen, mielőtt a csatornákat használó szolgáltatás végrehajtásra kerül. Ez megoldható egyszerűen úgy, hogy a modul egy adott időn keresztül nem végez feldolgozást, az idő pedig kiszámítható az átadott információmennyiség alapján.

Azonban - hacsak nem megfelelő a kivitelezés - az időkésleltetéseket a nem látható csatornákat kihasználó programok gyakran megkerülhetik. A kihasználó folyamatok például létrehozhatnak több küldő/fogadó halmazt. Miközben a TCB a késleltetési eljárások segítségével egyszerűen korlátozni tudja az egyes halmazokat, az összes halmaz összesítése kiadja a csatorna sávszélességét.

Jobb megoldás, ha egy adott TCB szolgáltatás biztosítja, hogy az időkésleltetések valamilyen módon a szolgáltatást használó összes folyamatra vonatkoznak.

Az időkésleltetések hasznosak lehetnek az elszigetelés és korlátozás során, de - mivel a rosszindulatú programok által viszonylag egyszerűen kivédhetők - gondos tervezést igényelnek.

Adatkorlátozások:

A nem látható csatorna sávszélessége nem csak az idő növelésével, hanem a visszaadott információmennyiség csökkentésével is csökkenthető. Az adatokat műveletsorozatok formájában visszaadó programok esetében gyakran egyszerűen megoldható, hogy ugyanazon az időkereten belül kevesebb vagy kisebb információcsomagot adjanak vissza.

Közelítő idő:

A nem látható csatornák kihasználását célzó számos eljárás igényli, hogy a kihasználó folyamatok a relatív vagy abszolút idő pontos mérésére módszert biztosítsanak. A csatornák néha korlátozhatók azáltal, hogy a folyamatnak nem biztosítja a pontos időmeghatározás lehetőségét.

Ugyan viszonylag egyszerű azt biztosítani, hogy az időinformációkat visszaadó TCB szolgáltatások megbecsüljék az időt, a folyamatok néha más módon is mérhetik az idő múlását, például saját utasításfeldolgozási idejük mérésével. A csatornák ilyen módon történő korlátozását tanácsos óvatosan használni.

Zajkeltők:

A legtöbb nem látható csatorna esetében a sávszélességet általában a csatornát kihasználó folyamatoktól eltérő folyamatok tevékenységei csökkentik, méghozzá esetenként igen jelentősen. Lehetséges, bár nem ajánlott, olyan megbízható programokat létrehozni, amelyek célja, hogy a csatornán folyamatosan egy adott szintű tevékenységet biztosítsanak. Az ilyen programokat zajkeltőknek hívjuk.

Ugyan a zajkeltők alkalmazása elméletileg vonzó lehet, a zajkeltőknek az esetek többségében nehéz lehet megállapítani, hogy mikor szükséges zajt kelteniük, illetve mikor nem. Ennek következtében az eljárás a nem látható csatornák korlátozására nem ajánlott.

U-T-U láncok:

Előfordulhatnak olyan helyzetek, amikor egy megbízhatatlan folyamat (**U1**) meghív egy megbízható folyamatot (**T**), amely ezt követően meghív egy másik megbízhatatlan folyamatot (**U2**), amely az **U1** folyamattól eltérő címkén fut. Az **U1** és az **U2** eltérő MAC címkéken futó megbízhatatlan folyamatokat képviselnek, és - mivel az egyik a másik leszármazottja - ez potenciálisan különleges nem látható csatorna létrejöttéhez vezethet. (Valójában a T és U lehetnek megbízható és/vagy megbízhatatlan folyamatok szekvenciái.) Az ilyen helyzeteket nevezzük U-T-U láncoknak.

A megbízható folyamatok biztosítják, hogy a két megbízhatatlan folyamat között átadott információk nem sértik az alapvető MAC elvet, amely egyaránt magában foglalja a nem megengedett közvetlen adatműveletek és a nem látható csatornák kizárását. Tanácsos az alábbiakat szem előtt tartani:

- A fájlleírók nem hagyhatók nyitva akkor, ha az **U2** nem nyithatta meg a fájlt abban az írási/olvasási módban, amelyben a fájl megnyitásra került
- A környezeti változókat ki kell üríteni akkor, ha az **U2** címkéje az U1 címkével szemben nem dominál
- Az **U1** felől az **U2** felé átadott munkakönyvtár nem látható csatornát alkothat (bár valószínűleg kicsit) akkor, ha az **U2** címkéje az **U1** címkéjével szemben nem dominál. Hasonlóképpen, számos, a leszármazott folyamatok által automatikusan örökölt folyamatparaméter nem látható csatornát alkothat.

Lehetséges az U-T-U láncok megfelelő kezelése (tehát úgy, hogy a nem látható csatornák elégségesen korlátozásra kerüljenek). Azonban ezt nehéz biztosítani, tehát tanácsos az U-T-U láncok használatát általában elkerülni. Fontos azonban megjegyezni, hogy a problémát az okozza, hogy az U2 nem megbízható. Ha megbízható, de nem privilegizált, akkor probléma nem áll fenn.

Nem látható csatorna - példák:

A rendszerprogramozó által létrehozott modulokban potenciálisan az alábbi nem látható csatornák fordulhatnak elő.

Nyomtatási szolgáltatás nem látható csatorna - példa:

Az alábbiakban egy nyomtatási szolgáltatás nem látható csatorna példáját mutatjuk be.

A megbízható sornyomtató szolgáltatások helyesen megjelölik az összes benyújtott feladatot a kérelmező folyamat MAC címkéjével, majd a címkét a sorba állított feladattal megőrzik és a későbbiek folyamán, a nyomtatáskor felhasználják. A viszonylagosan hosszú névvel rendelkező feladatok használata engedélyezett.

Az állapotprogramok segítségével a felhasználók megtekinthetik az összes várakozási sorban álló feladatot, beleértve a felhasználó által a feladathoz rendelt feladatnevet, a feladat címkéjétől függetlenül. A csatorna nem látható csatornaként is használható, hiszen a küldő folyamat ezt követően létrehozhat feladatokat, amelyek neve a felhasználó nevében működő fogadók felé rejtetten továbbítandó adatokat tartalmaz.

Megjegyzés: A nem látható kihasználás egyetlen feltétele, hogy a fogadó címkéje ne dominál jon a küldő címkéjével szemben, illetve hogy a küldő és fogadó egyaránt megbízhatatlan legyen. Gyakran mind a küldő, mind a fogadó ugyanannak a felhasználónak a nevében kerül felhasználásra.

A csatorna lezárása úgy kivitelezhető, hogy a felhasználó csak a felhasználó MAC címkéje által dominált feladatokat tekinthesse meg. Ezzel kényszeríti a fogadó MAC címkéjét, hogy a küldő MAC címkéjét dominálja, és a csatorna csak legális kiemelésre használható. Figyelmességből az állapotprogram a felhasználót "egyéb feladatok léteznek" üzenet formájában tájékoztathatja akkor, ha nem dominált feladatok is léteznek. Ez sokkal kisebb csatornát képvisel, illetve létezésének jól meghatározott működési oka van.

Megjegyzés: A magasabb szintű feladatok észlelésének megfigyelése szintén hasznos lehet, hiszen a szokásos üzemeltetés során az észlelések valószínűleg ritkák lesznek.

Ez az olyan nem látható csatornák általános példája, ahol a többszintű adatobjektumok (esetünkben sorba állított nyomtatási feladatok) elérhetők a különböző MAC címkéken futó folyamatok számára. A csatorna ténylegesen eltávolításra kerül azáltal, hogy az objektum MAC címkéjét a névre is alkalmazza. Nem látható információkat - a név mellett - egyéb attribútumok is tartalmazhatnak.

Erőforrástárak - példa:

Amikor egy megbízható program egy megbízhatatlan kliens számára szolgáltatást hajt végre, akkor a megbízható program egy adott típusú erőforrást (például puffert) lefoglal a különböző MAC címkéken futó folyamatok között megosztott erőforrástárból.

Ennek nem látható csatornakénti egyik felhasználási módja annak elrendezése, hogy a küldő és fogadó számára - egy kivételével - az összes erőforrás lefoglalásra kerüljön. Ezt potenciálisan elvégezhetik az eltérő vagy különböző MAC címkéken, illetve eltérő vagy különböző felhasználói azonosítók alatt futó egyéb programok is. A küldő ezt követően az egyetlen maradék erőforrást lefoglalttá vagy nem foglalttá teszi, majd a fogadó ezt észleli azáltal, hogy szintén megkísérli az erőforrás lefoglalását.

Ennek klasszikus példája az erőforrás-csatorna. Az erőforrás-csatorna a címkealapú erőforrástárak lefoglalásával korlátozható, a fentiekben leírt módon. Az erőforrás-csatornát ezen kívül a megfigyelés is észlelheti.

Adatbázisok - példa:

A megbízható adatbázisrendszerek lehetővé teszik, hogy a felhasználói programok az adatokat többszintű adatbázisokban helyezzék el. A közvetlen hozzáférés megfelelő vezérléséért az alapszintű MAC megszorítások felelősek.

Azonban az adatbázis-bejegyzés elhelyezéséhez szükséges idő jelentősen függ az adatbázis teljes méretétől. Ennek következtében a küldő az adatbázis méretét a bejegyzések hozzáadásával és eltávolításával befolyásolhatja, illetve a fogadó az adatbázis méretét egyszerűen megállapíthatja az adatbázis-bejegyzés elhelyezéséhez szükséges idő mérésével. A csatorna valószínűleg alacsony sávszélességgel rendelkezik, hacsak az adatbázis-hozzáférés nem nagyon hatékony.

A csatorna korlátozásához garantált minimális hozzáférési idő szabható meg. Az időkésleltetés lehet pszeudó-véletlen, hogy ezáltal átlagosan az időveszteség alacsonyabb legyen. Ennek ellenére továbbra is időkésleltetési sémáról beszélünk, tehát a kivitelezés során óvatosan kell eljárni.

Az összes hozzáférés egyszerű megfigyelése valószínűleg nem hatékony, hiszen a csatorna kihasználása nehezen észlelhető az adatbázis számos nem rosszindulatú felhasználása között.

Programozási példák:

A szakasz számos megbízható programozási példát tartalmaz

Megbízható program jogosultságellenőrzése - példa:

Az alábbi moduláris rutin segítségével a megbízható program ellenőrizheti, hogy a hívó folyamat rendelkezik-e egy adott jogosultsággal vagy sem.

```
#include <sys/priv.h>
#include <sys/secattr.h>

int
priv_check (int priv)
{
  /* a folyamat biztonsági attribútumai */
  secattr_t secattr;

/* a hívó folyamat biztonsági attribútumainak megszerzése */
  if ( sec_getpsec(-1, &secattr;) != 0 )
  {
    return (-1);
/* hiba a folyamat cred struktúrájának megszerzése során */
  }

  /*
   * annak visszaadása, hogy a megadott jogosultság a hívó folyamat
   * maximális jogosultsághalmazának eleme-e vagy sem
   */
  return privbit_test(secattr.sc_maxpriv, priv);
}
```

Tényleges érzékenységi címke módosítása - példa:

Az alábbi program az aktuális folyamat érzékenységi címkéjét a rendszer magas értékre módosítja.

A program belső jogosultsághalmazának az alábbi jogosultságokat kell tartalmaznia:

- **PV_LAB_LEF**
- **PV_LAB_SLUG**
- **PV_LAB_SL_SELF**

```
#include <stdio.h>
#include <mls/mls.h>
#include <unistd.h>
#include <sys/secattr.h>
#include <userpriv.h>
#include <sys/mac.h>
#include <sys/secconf.h>

#define SUCCESS 0
#define ERROR 1

int
main()
{
 sl_t sl_syshi; /* Rendszer magas SL */
 secattr_t attr;
 char *clBuffer = NULL;

 /*
  * A rendszer magas és alacsony SL címkéjének megszerzése.
  */
 if ((sec_getsyslab(NULL, &sl_syshi, NULL, NULL)) != 0 ) {
  fprintf (stderr, "Call to sec_getsyslab failed.\n");
  exit(ERROR);
 }

 /*
  * A rendszer alapértelmezett címkeadatbázisának eléréséhez
```

```
 * a folyamatot az initlabeldb() függvénnyel inicializáljuk.
 */
priv_raise(PV_LAB_LEF , -1);
if (initlabeldb(NULL) != 0) {
 fprintf(stderr, "A címkekódolási adatbázis nem olvasható.\n");
 exit(ERROR);
}
priv_remove(PV_LAB_LEF, -1);

/*
 * A folyamat jogosultsági tartományának és hatályos SL címkéjének megszerzése.
 */
priv_raise(PV_LAB_SLUG, PV_LAB_SL_SELF, -1);
if (sec_getpsec(-1, &attr) != 0) {
 fprintf (stderr, "Hiba a program Trusted AIX biztonsági attribútumainak megszerzése során.\n");
 exit(ERROR);
}

/* a folyamathoz létrehozható maximális SL címkehossz malloc-ja */
if((clBuffer = (char *) malloc(maxlen_cl())) == NULL) {
 perror("malloc");
 exit(ERROR);
}
/* Bináris hatályos SL átalakítása felhasználó által olvashatóra */
if (clbtohr(clBuffer, &attr.sc_sl, HR_LONG) != 0) {
 fprintf(stderr, "Az SL nem alakítható át felhasználó által olvasható formátumúra.\n");
 exit(ERROR);
}
printf("A program kezdeti hatályos SL címkéje = %s.\n",clBuffer);

/*
 * A folyamat hatályos SL címkéjének beállítása a rendszer magas értékre.
 * Előfordulhat, hogy a folyamat maximális SL címkéje nem a rendszer
 * magas érték, ezért ezt is beállítjuk a rendszer magas értékre.
 */
attr.sc_sl = sl_syshi;
attr.sc_sl_cl_max = sl_syshi;

if (sec_setplab(-1, &attr.sc_sl, NULL, &attr.sc_sl_cl_max,
  NULL, NULL, NULL) != 0) {
 fprintf (stderr, "Hiba a program hatályos SL címkéjének beállítása során.\n");
 exit(ERROR);
}

priv_lower(PV_LAB_SLUG, PV_LAB_SL_SELF, -1);

if (sec_getpsec(-1, &attr) != 0) {
 fprintf (stderr, "Hiba a program Trusted AIX biztonsági attribútumainak megszerzése során.\n");
 exit(ERROR);
}

/* Bináris hatályos SL átalakítása felhasználó által olvashatóra */
if (clbtohr(clBuffer, &attr.sc_sl, HR_LONG) != 0) {
 fprintf(stderr, "Az SL nem alakítható át felhasználó által olvasható formátumúra.\n");
 exit(ERROR);
}
printf("Program módosított hatályos SL címkéje = %s.\n",clBuffer);
return(SUCCESS);
}
```

Érzékenységicímke-besorolások beállítása és érzékenységi címkék összehasonlítása - példa:

Az alábbi példa bemutatja az érzékenységi címkék besorolásának beállítását, illetve a függvénytár rutinjainak felhasználását az érzékenységi címkék közötti összehasonlításokra.

A program proxy jogosultsághalmazának, illetve a hívó folyamat maximális jogosultsághalmazának tartalmaznia kell a **PV_LAB_LEF** jogosultságot.

```
#include <stdio.h>
#include <mls/mls.h>
#include <userpriv.h>
#include <errno.h>

#define SUCCESS 0
#define ERROR 1
int
main (int argc, char **argv)
{
/* Érzékenységi címkék */
sl_t sl1, sl2;

/* a címkék nevét tartalmazó karaktersorozatok */
char *slBuffer1 = NULL;
char *slBuffer2 = NULL;

if (argc != 3) {
fprintf(stderr, "Használat: compare sl1 sl2\n");
exit(ERROR);
}
/*
* A rendszer alapértelmezett címkeadatbázisának eléréséhez
* a folyamatot az initlabeldb() függvénnyel inicializáljuk.
*/
priv_raise(PV_LAB_LEF , -1);
if (initlabeldb(NULL) != 0) {
fprintf(stderr, "A címkekódolási adatbázis nem olvasható.\n");
exit(ERROR);
}
priv_remove(PV_LAB_LEF, -1);

/* Az átadott SL átalakítása bináris formátumúra */
if (slhrtob(&sl1, argv[1]) != 0) {
fprintf(stderr, "A(z) %s bináris formátumúra nem alakítható át.\n", argv[1]);
exit(ERROR);
}
if (slhrtob(&sl2, argv[2]) != 0) {
fprintf(stderr, "A(z) %s bináris formátumúra nem alakítható át.\n", argv[2]);
exit(ERROR);
}

/* a létrehozható maximális SL címkehossz malloc-ja */
slBuffer1 = (char *) malloc(maxlen_sl());
slBuffer2 = (char *) malloc(maxlen_sl());

if ((slBuffer1 == NULL) || (slBuffer2 == NULL)) {
perror("malloc");
exit(ERROR);
}

/*
* A címke visszaalakítása felhasználó által olvasható (hosszú) formátumra.
* A lépés nem szükséges, csupán a slbtohr() API felhasználását
* mutatja be.
*/
if (slbtohr(slBuffer1, &sl1, HR_LONG) != 0) {
fprintf(stderr, "Nem alakítható át bináris felhasználó által olvasható formátumúra.\n");
exit(ERROR);
}

if (slbtohr(slBuffer2, &sl2, HR_LONG) != 0) {
fprintf(stderr, "Nem alakítható át bináris felhasználó által olvasható formátumúra.\n");
exit(ERROR);
```

```
}

/*
* Az sl_cmp() segítségével hasonlítsa össze a két címke dominanciáját.
*/
if (sl_cmp(&sl1, &sl2) == LAB_SAME) {
printf("A(z) (%s) címke megegyezik a(z) (%s) címkével.\n",
slBuffer1, slBuffer2);
}
else if (sl_cmp(&sl1, &sl2) == LAB_DOM) {
printf("A(z) (%s) címke a(z) (%s) címkét meghatározza.\n",
slBuffer1, slBuffer2);
}
else if (sl_cmp(&sl2, &sl1) == LAB_DOM) {
printf("A(z) (%s) címke a(z) (%s) címkét meghatározza.\n",
slBuffer2, slBuffer1);
}
else {
printf("A két címke különálló.\n");
}

return (SUCCESS);
}
```

Megfigyelési információk beállítása - példa:

A program lekéri és beállítja a megfigyelési információkat.

A program belső jogosultsághalmazának az alábbi jogosultságokat kell tartalmaznia:

- **PV_AU_ADMIN**
- **PV_DAC_GID**

```
#include <sys/types.h>
#include <sys/priv.h>
#include <sys/audit.h>

char buf[1024];
int main(int argc, char *argv[])
{
 int rc, len, p;
 /* A folyamat megfigyelés-előkijelölési maszkjának megszerzése */
 priv_raise(PV_AU_ADMIN, -1);
 rc = auditproc(0, AUDIT_QEVENTS, buf, sizeof (buf));
 priv_lower(PV_AU_ADMIN, -1);
 if (rc)
  fprintf(stderr, "A megfigyelési információk megszerzése meghiúsult\n");
        /* *Adja hozzá a `kernel megfigyelési osztályt az előkijelölési maszkhoz */
 p = 0;
 while ((len = strlen(&buf;[p])) > 0)
  p += len +1;
        strncat(&buf;[p], "kernel", (sizeof(buf)-p-1));
 p += strlen("kernel") + 2;
 buf[p] = 0;
 priv_raise(PV_AU_ADMIN, -1);
 rc = auditproc(0, AUDIT_EVENTS, buf, p);

 priv_lower(PV_AU_ADMIN, -1);
 if (rc)
  fprintf(stderr, "A megfigyelési információk beállítása meghiúsult\n");
 /* *Állítsa be a folyamat GID azonosítóját úgy, hogy megfigyelési rekordot állítson elő */
 priv_raise(PV_DAC_GID, -1);
 rc = setgid(129);
 priv_lower(PV_DAC_GID, -1);
```

```
 if (rc)
  fprintf(stderr, "A setgid meghiúsult\n");
 exit(0);
}
```

Kliens példa:

A program a szerver felé két üzenetet küld, egyiket a szabványos **write**, a másikat az **ewrite** rutin segítségével.

A biztonságos üzenet SECRET címkével kerül továbbításra. Fontos megjegyezni, hogy a **write** hívással továbbított nem biztonságos üzenetek megkapják az alapértelmezett biztonsági attribútum halmazt, amely a netrule segítségével állítható be.

A program belső jogosultsághalmazának az alábbi jogosultságokat kell tartalmaznia:

- **PV_LAB_LEF**
- **PV_MAC_CL**
- **PV_LAB_SLUG_STR**

```
#include  <sys/mac.h>
#include  <sys/socket.h>
#include  <netinet/in.h>
#include  <sys/priv.h>
#include  <sys/secattr.h>
#include  <errno.h>
#include  <stdio.h>
#define SECURE 1
int
main(int  argc,  char  *argv[])

{
 int  sockfd;
 int  uid,  gid;
 char  buf[BUFSIZ];

 struct  sockaddr_in  serv_addr;

#ifdef  SECURE
 int  l_init_result  =  0;

 int  ewrite_result  =  0;

 sec_labels_t  seclab;

#endif  /*SECURE*/


 uid  =  getuid();

 gid  =  getgid();


 if  (  argc  !=  3  )

 {
  fprintf(stderr,  "Használat: %s:  ADDR  PORT\n",  argv[0]);

  exit(1);
 }
#ifdef  SECURE
 /*
  * *  Hozzáférés a címkekódolási adatbázishoz
  *
  *     */
```

```
 priv_raise(PV_LAB_LEF,-1);
 l_init_result  =  initlabeldb(NULL);
 if  (  priv_remove(PV_LAB_LEF,  -1)  !=  0  )
 {
  fprintf(stderr,  "Jogosultság hiba\n");
  exit(1);
 }
 if  (  l_init_result  !=  0  )
 {
  fprintf(stderr, "A címkekódolási adatbázis nem olvasható\n");
  exit(0);
 }
#endif  /*SECURE*/
  /*
   * *  A "serv_addr" struktúrát helyettesítse annak a szervernek
   * *
   * *  a címével, amelyhez csatlakozni kíván.
   * */
 memset  ((char  *)  &serv_addr;,  '\0',  sizeof(serv_addr));
 serv_addr.sin_family  =  AF_INET;
 serv_addr.sin_addr.s_addr  =  inet_addr(argv[1]);
 serv_addr.sin_port  =  htons(atoi(argv[2]));
 /*  Nyisson meg egy TCP  socketet (Internet  adatfolyamsocketet).  */
 if  (  (sockfd  =  socket(AF_INET,  SOCK_STREAM,  0))  <  0)
 {
  perror("tcpkliens:  ");
  fprintf(stderr,  "kliens:  Az adatfolyamsocket nem nyitható meg\n");
  exit(0);
 }
 if  (  connect(sockfd,  (struct  sockaddr  *)  &serv_addr;,
    sizeof(serv_addr))  <  0  )
 {
  perror("tcpkliens:  ");
  fprintf(stderr,  "kliens:  A szerverhez nem lehet csatlakozni\n");
  exit(0);
 }
 /*
  * *  Szabványos írás küldése a szerver felé, amelyhez hozzárendelésre
  * *  kerülnek az alapértelmezett biztonsági attribútumok
  * */
 strcpy(buf,  "Rendelkezik az alapértelmezett biztonsági attribútumokkal.\n");
 if  (  write(sockfd,  buf,  strlen(buf)+1)  ==  -1  )
 {
  perror("tcpkliens:  ");
  fprintf(stderr,  "írási hiba\n");
 }
#ifdef  SECURE
  strcpy(buf,  "Az üzenet biztonsági címkéje SECRET\n");
 /*  Az SL  és a CL címkék beállítása */
 slhrtob(&seclab.sl;, "SECRET");
 slhrtob(&seclab.sl_cl_min;,  "SECRET");
 slhrtob(&seclab.sl_cl_max;,  "SECRET  A  B");
 seclab.sl.sl_format  =  STDSL_FORMAT;
 seclab.sl_cl_min.sl_format  =  STDSL_FORMAT;
 seclab.sl_cl_max.sl_format  =  STDSL_FORMAT;
 /*  Az ewrite  hívás PV_MAC_CL  és PV_LAB_SLUG_STR  jogosultságot igényel */
 priv_raise(PV_MAC_CL,PV_LAB_SLUG_STR,-1);
 ewrite_result  =  ewrite(sockfd,  buf,strlen(buf)+1,  &seclab;);
 priv_lower(PV_MAC_CL,PV_LAB_SLUG_STR,-1);

 if  (ewrite_result  ==  -1)
 {
  perror("tcpkliens-hívás");
  fprintf(stderr,  "ewrite hiba\n");
 }
 fflush(stderr);
```

```
#endif  /*SECURE*/
 fprintf(stderr,  "kilépés .....  \n");
 sleep(3);
 close(sockfd);
 exit(0);
}
```

Szerver példa:

Az alábbi program szerverként üzemel, és az **eread** rutin segítségével fogadja a portjára érkező üzeneteket. Az üzenetek sikeres fogadását követően a program kimenetként megjeleníti az üzenet biztonsági attribútumait.

A program belső jogosultsághalmazának az alábbi jogosultságokat kell tartalmaznia (az FSF_EPS biztonsági kapcsolók hozzárendelése nélkül):

- **PV_LAB_LEF**
- **PV_MAC_CL**
- **PV_MAC_R_STR**

```
#include  <sys/mac.h>
#include  <sys/socket.h>
#include  <sys/priv.h>
#include  <sys/secattr.h>
#include  <sys/stropts.h>
#include  <netinet/in.h>
#include  <errno.h>
#include  <stropts.h>
#include  <unistd.h>
#include  <stdio.h>
#include  <mls/mls.h>
#define  MAX_HR_LABEL_LEN 2048
#define  SECURE 1
int
main(int  argc,  char  *argv[])
{
 pid_t  childpid;
 uint  clilen;
 int  sockfd,  newsockfd;
 struct  sockaddr_in  cli_addr,  serv_addr;


#ifdef  SECURE
 int  l_init_result;
 char  label_str[MAX_HR_LABEL_LEN];
 sec_labels_t  seclab;
#endif  /*  SECURE  */
 if  (  argc  !=  2  )
 {
  fprintf(stderr,  "Használat:%s  PORT\n",  argv[0]);
  exit(1);
 }
#ifdef  SECURE
 priv_raise(PV_LAB_LEF,  -1);
 l_init_result  =  initlabeldb(NULL);
 if  (priv_remove(PV_LAB_LEF,  -1)  !=  0)
 {
  fprintf(stderr,  "Jogosultság hiba\n");
  exit(1);
 }

 if  (l_init_result  !=  0)
 {
  fprintf(stderr, "A címkekódolási adatbázis nem olvasható\n");
  exit(1);
 }
#endif  /*  SECURE  */
```

```
/*  Nyisson meg egy TCP  socketet (Internet  adatfolyamsocketet).  */
if  (  (sockfd  =  socket(AF_INET,  SOCK_STREAM,  0))  <  0  )
{
 perror("tcpszerver:  ");
 fprintf(stderr,  "szerver:  Az adatfolyamsocket nem nyitható meg\n");
 exit(1);
}
/*A helyi cím kötése, hogy a szerver nekünk ne küldjön adatokat*/
memset((char  *)  &serv_addr;,  '\0',  sizeof(serv_addr));
serv_addr.sin_family  =  AF_INET;
serv_addr.sin_addr.s_addr  =  htonl(INADDR_ANY);
serv_addr.sin_port  =  htons(atoi(argv[1]));
if  (  bind(sockfd,  (struct  sockaddr  *)  &  serv_addr,
   sizeof(serv_addr))  <  0  )
{
 perror("tcpszerver:  ");
 fprintf(stderr,  "szerver:  A helyi cím nem köthető\n");
 exit(0);
}
listen(sockfd,  5);
for  (;;)
{
 /*
  * *  Kliensfolyamat csatlakozására vár.
  * */
 fprintf(stdout,  "Klienskapcsolatra vár...\n");
 clilen  =  sizeof(cli_addr);
 newsockfd  =  eaccept(sockfd,  (struct  sockaddr  *)  &      cli_addr,
   &clilen;,  &seclab;);
 if  (  newsockfd  <  0  )
 {
  perror("tcpszerver:  ");
  fprintf(stderr,  "szerver:  elfogadási hiba\n");
 }
 /*  SL nyomtatása */
 if  (  slbtohr(label_str,  &seclab.sl;,  HR_SHORT)  !=  0  )
 {
  fprintf(stderr,"hiba az sl karaktersorozattá alakítása során\n");
 }
 else
 {
  fprintf(stdout,  "sl  =  %s.\n",label_str);
 }
 /*  MIN CLEARANCE nyomtatása */
 if  (  slbtohr(label_str,  &seclab.sl_cl_min;,  HR_SHORT)  !=  0  )
 {
  fprintf(stderr,"hiba a minimális engedély karaktersorozattá alakítása során\n");
 }
 else
 {
  fprintf(stdout,  "sl_cl_min  =  %s.\n",label_str);
 }

 /*  MAX CLEARANCE nyomtatása */
 if  (  slbtohr(label_str,  &seclab.sl_cl_max;,  HR_SHORT)  !=  0  )
 {
  fprintf(stderr,"hiba a maximális engedély karaktersorozattá alakítása során\n");
 }
 else
 {
  fprintf(stdout,  "sl_cl_max  =  %s.\n",label_str);
 }
 if  (  (childpid  =  fork())  <  0  )
 {
  perror("tcpszerver:  ");
  fprintf(stderr,  "szerver:  fork hiba\n");
  exit(0);
```

```
  }
  else  if  (  childpid  ==  0  )  /*  utód folyamat  */
  {
   int  i,  j;
   char  buf[BUFSIZ];
#ifdef  SECURE
  sec_labels_t  e_seclab;
#endif  /*  SECURE  */
  close(sockfd);
  for  (;;)
  {
      int  ret,  flag;
      struct  strbuf  ctstr,  dtstr;
      char  ctbuf[2048],  dtbuf[2048];
      ctstr.maxlen=2048;
      ctstr.buf  =  ctbuf;
      dtstr.maxlen=2048;
      dtstr.buf  =  dtbuf;
#ifdef  SECURE
      fprintf(stdout,  "Calling  eread\n");
      priv_raise(PV_MAC_CL,PV_MAC_R_STR,-1);
      ret  =  eread(newsockfd,  buf,  sizeof(buf),&e_seclab;);
      priv_lower(PV_MAC_CL,PV_MAC_R_STR,-1);
      if  (  ret  <  1  )
     {
         if  (  ret  ==  -1  )
   fprintf(stderr,  "eread  error\n");
        else
   fprintf(stderr,  "eread  no  data\n");
   close(newsockfd);
   exit(ret);
      }
      fprintf(stdout,  "\n%s",  buf);
      fprintf(stdout,  "\n");
      /*  SL nyomtatása */
      if  (  slbtohr(label_str,  &e_seclab.sl;,  HR_SHORT)  !=  0  )
      {
   fprintf(stderr,"hiba az sl karaktersorozattá alakítása során\n");
       }
      else
      {
   fprintf(stdout,  "sl  =  %s.\n",label_str);
       }
       /*  MIN CLEARANCE nyomtatása */
       if  (  slbtohr(label_str,&e_seclab.sl_cl_min;,HR_SHORT)!=  0)
      {
   fprintf(stderr,"probléma a min CL karaktersorozattá alakítása során\n");
       }
      else
      {
   fprintf(stdout,  "sl_cl_min  =  %s.\n",label_str);
       }
       /*  MAX CLEARANCE nyomtatása */
      if  (  slbtohr(label_str,&e_seclab.sl_cl_max;,HR_SHORT)  !=0)
      {
   fprintf(stderr,"probléma a max CL karaktersorozattá alakítása során\n");
       }
       else
              {
   fprintf(stdout,  "sl_cl_max  =  %s.\n",label_str);
        }
        fflush(stdout);
#else  /*  NOT  SECURE  */
        fprintf(stdout,  "Olvasás meghívása\n");
        if  (read(newsockfd,  buf,  sizeof(buf))  <  1)
      {
   if  (ret  ==  -1)
```

```
      fprintf(stderr,  "olvasási hiba\n");
   else
      fprintf(stderr,  "nincs beolvasott adat\n");
   close(newsockfd);
   exit(ret);
        }
        fprintf(stdout,  "%s\n",  buf);
        fflush(stdout);
#endif  /*  NOT  SECURE  */
              }
 }
 /*  szülőfolyamat */
 close(newsockfd);
        }
}
```

Trusted AIX felhasználói- és port biztonsági attribútumok:

A felhasználói és port biztonsági attribútumok segítségével kérhető le a felhasználók és portok engedélyattribútumai, illetve hasonlíthatók össze a felhasználók engedélyattribútumai a port attribútumaival.

A következő kiegészítő attribútumok vannak meghatározva a Trusted AIX **usersec.h** fájljában.

S_MINSL
A felhasználó minimális érzékenységi engedélycímkéje. Típus: SEC_CHAR

S_MAXSL
A felhasználó maximális érzékenységi engedélycímkéje. Típus: SEC_CHAR

S_DEFSL
A felhasználó alapértelmezett érzékenységi címkéje. Típus: SEC_CHAR

S_MINTL
A felhasználó minimális integritás-engedélycímkéje. Típus: SEC_CHAR.

S_MAXTL
A felhasználó maximális integritás-engedélycímkéje. Típus: SEC_CHAR.

S_DEFTL
A felhasználó alapértelmezett integritás címkéje. Típus: SEC_CHAR

A portok esetében az alábbi attribútumok érvényesek:

S_MINSL
A porthoz rendelt minimális érzékenységi címke. Típus: SEC_CHAR.

S_MAXSL
A porthoz rendelt maximális érzékenységi címke. Típus: SEC_CHAR

S_TL A porthoz rendelt integritás címke. Típus: SEC_CHAR

Az alábbi példa meghatározza, hogy a felhasználó bejelentkezhet-e a megadott porton.

```
#include <mls/mls.h>
#include <usersec.h>
#include <stdio.h>
#include <errno.h>

struct userlabels {
 sl_t minsl;
 sl_t maxsl;
 sl_t defsl;
 tl_t mintl;
 tl_t maxtl;
 tl_t deftl;
};
```

```
 struct portlabels {
  sl_t minsl;
  sl_t maxsl;
  tl_t tl;
 };

 void getuserlabels(char * username, struct userlabels *usrlab);
 void getportlabels (char * portname, struct portlabels *portlab);
 void displayuseraccess (char * username, struct userlabels *usrlab,
      struct portlabels *portlab);

 int
 main (int argc, char **argv)
 {

  struct userlabels usrlab;
  struct portlabels portlab;
  char *username = NULL;
  char *portname = NULL;


  if (argc != 3 ) {
   fprintf (stderr, "Használat: %s <felhasználónév> <portnév>\n",
argv[0]);
   exit(1);
  }
  username = argv[1];
  portname = argv[2];

  initlabeldb(NULL);
  getuserlabels(username, &usrlab;);
  getportlabels(portname, &portlab;);
  displayuseraccess(username , &usrlab;, &portlab;);
  endlabeldb();
 }

 void getuserlabels(char *username,  struct userlabels *userlab)
 {

  dbattr_t attributes[6];
  memset (attributes, 0, sizeof(attributes));

  attributes[0].attr_name = S_MINSL;
  attributes[0].attr_type = SEC_CHAR;

  attributes[1].attr_name = S_MAXSL;
  attributes[1].attr_type = SEC_CHAR;

  attributes[2].attr_name = S_DEFSL;
  attributes[2].attr_type = SEC_CHAR;

  attributes[3].attr_name = S_MINTL;
  attributes[3].attr_type = SEC_CHAR;

  attributes[4].attr_name = S_MAXTL;
  attributes[4].attr_type = SEC_CHAR;

  attributes[5].attr_name = S_DEFTL;
  attributes[5].attr_type = SEC_CHAR;

  if (getuserattrs(username, attributes, 6)) {
   fprintf(stderr,
     "Hiba a(z) %s felhasználó attribútumainak lekérése során\n", username);
   exit (1);
  }
```

```
 if (clhrtob (&(userlab->minsl), attributes[0].attr_char)) {
  fprintf(stderr, "minsl átalakítási hiba\n");
  exit (1);
 }

 if (clhrtob(&(userlab->maxsl), attributes[1].attr_char)) {
  fprintf(stderr, "maxsl átalakítási hiba\n");
  exit (1);
 }

 if (clhrtob(&(userlab->defsl), attributes[2].attr_char)) {
  fprintf(stderr, "defsl átalakítási hiba\n");
  exit (1);
 }

 if (tlhrtob(&(userlab->mintl), attributes[3].attr_char)) {
  fprintf(stderr, "mintl átalakítási hiba\n");
  exit (1);
 }

 if (tlhrtob(&(userlab->maxtl), attributes[4].attr_char)) {
  fprintf(stderr, "maxtl átalakítási hiba\n");
  exit (1);
 }

 if (tlhrtob(&(userlab->deftl), attributes[5].attr_char)) {
  fprintf(stderr, "deftl átalakítási hiba\n");
  exit (1);
 }

 printf("A(z) %s felhasználó a következő engedélyértékekkel rendelkezik:\n", username);
 printf("minsl:%s\n", attributes[0].attr_char);
 printf("maxsl:%s\n", attributes[1].attr_char);
 printf("defsl:%s\n", attributes[2].attr_char);
 printf("mintl:%s\n", attributes[3].attr_char);
 printf("maxtl:%s\n", attributes[4].attr_char);
 printf("deftl:%s\n", attributes[5].attr_char);

 return;
}

void getportlabels(char *portname, struct portlabels *portlab)
{
 int rc =0;
 char *val = NULL;
 if ( (rc = getportattr(portname,S_MINSL,(char*)&val;, SEC_CHAR)) != 0 ) {
  perror ("Hiba a portattribútumok lekérése során");
  exit(1);
 }

 if (slhrtob(&(portlab->minsl), val)) {
  fprintf(stderr, "port minsl átalakítási hiba\n");
  exit (1);
 }

 if ( (rc = getportattr(portname,S_MAXSL, (char*)&val;, SEC_CHAR)) != 0 ) {
  perror ("Hiba a portattribútumok lekérése során");
  exit(1);
 }

 if (slhrtob(&(portlab->maxsl), val)) {
  fprintf(stderr, "port maxsl átalakítási hiba\n");
  exit (1);
 }

 if ( (rc = getportattr(portname,S_TL, (char*)&val;, SEC_CHAR)) != 0 ) {
  perror ("Hiba a portattribútumok lekérése során");
```

```
  }

  if (tlhrtob(&(portlab->tl), val)) {
   fprintf(stderr, "port tl átalakítási hiba\n");
   exit (1);
  }

  return;
 }

 void displayuseraccess (char *username, struct userlabels *usrlab, struct portlabels *portlab)
 {
  CMP_RES_T cmpres;
  cmpres = sl_cmp(&(usrlab->defsl), &(portlab->minsl));
  if (cmpres != LAB_DOM && cmpres != LAB_SAME) {
   printf("A felhasználó alapértelmezett SL címkéje nem dominál a tty minimális SL címkéjével szemben \n");
   exit(1);
  }

  cmpres = sl_cmp(&(portlab->maxsl), &(usrlab->defsl));
  if (cmpres != LAB_DOM && cmpres != LAB_SAME) {
   printf("A tty maximális SL címkéje a felhasználó alapértelmezett SL címkéjével szemben nem domoinál\n");
   exit(1);
  }

  cmpres = tl_cmp(&(portlab->tl), &(usrlab->deftl));
  if (cmpres != LAB_SAME) {
   printf("A felhasználó alapértelmezett TL címkéje nem egyezik meg a tty TTL címkéjével \n");
   exit(1);
  }

  printf("A felhasználó a megadott porton be tud jelentkezni\n");
  return;

}
```

Megbízható AIX rendszerhívások:

A kiegészítő Trusted AIX funkcionalitás kezeléséhez a rendszer rendszerhívásokat biztosít.

eaccept
Socketkapcsolat elfogadása

ebind A binds kiterjesztése a biztonsági attribútumok használatával

econnect
A socketkapcsolat kezdeményezésének kiterjesztése a biztonsági attribútumok használatával

eread Adatfolyam beolvasása, majd az üzenet biztonsági attribútumainak lekérése

ereadv Adatfolyam beolvasása, majd az üzenet biztonsági attribútumainak lekérése

erecv A recv, recvfrom és recvmsg kiterjesztése a biztonsági attribútumok használatával

erecvfrom
A recv, recvfrom és recvmsg kiterjesztése a biztonsági attribútumok használatával

erecvmsg
A recv, recvfrom és recvmsg kiterjesztése a biztonsági attribútumok használatával

esend A send, sendto és sendmsg kiterjesztése a biztonsági attribútumok használatával

esendmsg
A send, sendto és sendmsg kiterjesztése a biztonsági attribútumok használatával

esendto
A send, sendto és sendmsg kiterjesztése a biztonsági attribútumok használatával

ewrite Adatfolyam írása, majd az üzenet biztonsági attribútumainak beállítása

ewritev
Adatfolyam írása, majd az üzenet biztonsági attribútumainak beállítása

sec_getmsgsec
Az üzenetsorok biztonsági attribútumainak lekérése

sec_getpsec
A folyamathoz tartozó biztonsági információk lekérése

sec_getrunmode
A kernel működési módjának lekérése

sec_getsecconf
Az aktuális biztonsági konfiguráció kapcsolóinak lekérése

sec_getsemsec
A szemaforok biztonsági attribútumainak lekérése

sec_getshmsec
Az osztott memóriaszegmensek biztonsági attribútumainak lekérése

sec_getsyslab
Az alapértelmezett rendszerérzékenységi címkék lekérése

sec_gettlibbufsize
A kernelben található függvénytárútvonal-bejegyzések lekérése

sec_gettlibpath
A kernelben található függvénytárútvonal-bejegyzések lekérése

pdmkdir
Particionált könyvtár vagy alkönyvtár létrehozása/beállítása/beállításának megszüntetése

sec_setauditrange
A rendszer globális megfigyelési címketartományának beállítása

sec_setplab
A megadott folyamat hatályos érzékenységi címkéjének, minimális érzékenységi engedélyének, maximális érzékenységi engedélyének illetve integritási címkéjének beállítása

setppdmode
A folyamat particionált könyvtár módjának (valós vagy virtuális) beállítása

setppriv
A folyamathoz tartozó jogosultsághalmazok beállítása

sec_setptlibmode
A folyamat TLIB módjának beállítása

sec_setrunmode
A kernel működési módjának beállítása

sec_setsecconf
A kernelbiztonsági konfiguráció kapcsolóinak beállítása

sec_setsemlab
A szemaforok biztonsági attribútumainak beállítása

sec_setshmlab
Az osztott memóriaszegmensek biztonsági attribútumainak beállítása

sec_setsyslab
Az alapértelmezett rendszer érzékenységi, információs és integritás címkéinek beállítása

AIX C függvénytár függvényei:

A kiegészítő Trusted AIX funkcionalitás kezeléséhez a rendszer szubrutinokat és makrókat biztosít.

accredrange
Meghatározza, hogy az érzékenységi címke az akkreditációs tartományon belüli-e.

clbtohr
Átalakítja a megadott bináris jogosultságcímkét felhasználó által olvasható formátumúra.

clhrtob
Átalakítja a megadott felhasználó által olvasható jogosultságcímkét bináris formátumúra.

getfsfbitindex, getfsfbitstring
A fájlbiztonsági kapcsoló karaktersorozatainak és indexeinek lekérdezésére szolgáló rutinok

getmax_sl, getmax_tl
Lekéri a maximális érzékenységi és integritás címkéket a címkekódolási fájlból.

getmin_sl, getmin_tl
Lekéri a minimális érzékenységi és integritás címkéket a címkekódolási fájlból.

getsecconfig, setsecconfig
A futtatási módok kernelbiztonsági kapcsolóinak lekérésére és beállítására szolgáló rutinok.

initlabeldb, endlabeldb
A címkeadatbázis inicializálására és lezárására szolgáló rutinok.

maxlen_sl, maxlen_cl, maxlen_tl
Lekéri a felhasználó által olvasható címkék maximális hosszát az inicializált címkekódolási fájl alapján.

priv_isnull
Meghatározza, hogy a megadott jogosultsághalmaz tartalmaz-e beállított jogosultságokat

priv_lower
Jogosultsághalmaz-műveletek

priv_raise
Jogosultsághalmaz-műveletek

priv_remove
Jogosultsághalmaz-műveletek

priv_subset
Jogosultsághalmaz-műveletek

privbit_clr
Törli a megadott jogosultságot a megadott jogosultsághalmazból

priv_clrall
Törli az összes jogosultságot a megadott jogosultsághalmazból

priv_comb
Egyesíti az első két megadott jogosultsághalmazt, majd az eredményeket a harmadik megadott jogosultsághalmazban helyezi el

priv_copy
Átmásolja az első megadott jogosultsághalmazt a második megadott jogosultsághalmazba

priv_isnull
Meghatározza, hogy a megadott jogosultsághalmazban vannak-e jogosultságok beállítva

priv_mask
Kiszámítja az első két megadott jogosultsághalmaz metszetét, majd az eredményeket a harmadik megadott jogosultsághalmazban helyezi el

priv_rem
Eltávolítja a második megadott jogosultsághalmaz jogosultságait az első megadott jogosultsághalamzból, majd az eredményeket a harmadik megadott jogosultsághalmazban helyezi el

privbit_set
Beállítja a megadott jogosultságot a megadott jogosultsághalmazban

priv_setall
Beállítja az összes jogosultságot a megadott jogosultsághalmazban

priv_subset
Meghatározza, hogy az első megadott jogosultsághalmaz a második megadott jogosultsághalmaz részhalmaza-e

privbit_test
Ellenőrzi, hogy a megadott jogosultságot a megadott jogosultsághalmaz tartalmazza-e

slbtohr, clbtohr, tlbtohr
A bináris címkék felhasználó által olvasható formátumúra alakítására szolgáló rutinok.

slhrtob, clhrtob, tlhrtob
A felhasználó által olvasható címkék bináris formátumúra alakítására szolgáló rutinok.

sl_clr, tl_clr
A címkék alaphelyzetbe állítására szolgáló rutinok

sl_cmp, tl_cmp
Címkék összehasonlítására szolgáló rutinok

tl_cmp Integritás címkék összehasonlítása

Megbízható AIX jogosultságok

A Trusted AIX rendszeren az alábbi jogosultságok állnak rendelkezésre. Az alábbiakban megtalálja az egyes jogosultságok összegzését, leírását, illetve használatát. Bizonyos jogosultságok hierarchiát alkotnak. Ebben az esetben az egyik jogosultság egy másik jogosultsághoz tartozó minden jogot adományozhat.

A jogosultságok ellenőrzése során a rendszer először meghatározza, hogy a folyamat rendelkezik-e a szükséges legalacsonyabb jogosultsággal, majd megy felfele a hierarchiában az erősebb jogosultságok meglétének ellenőrzése érdekében. A **PV_AU_** jogosultsággal rendelkező folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultsággal rendelkező folyamat automatikusan rendelkezik az összes alább felsorolt jogosultsággal, a **PV_SU_** jogosultságok kivételével.

Megfigyelési jogosultságok:

A Trusted AIX rendszeren az alábbi megfigyelési jogosultságok állnak rendelkezésre. Az alábbiakban megtalálja az egyes jogosultságok összegezését, leírását, illetve használatát. Bizonyos jogosultságok hierarchiát alkotnak. Ebben az esetben az egyik jogosultság egy másik jogosultsághoz tartozó minden jogot adományozhat.

A jogosultságok ellenőrzése során a rendszer először meghatározza, hogy a folyamat rendelkezik-e a szükséges legalacsonyabb jogosultsággal, majd az ellenőrzést folytatja, felfelé a hierarchiában, az erőteljesebb jogosultságok felé. A **PV_AU_** jogosultsággal rendelkező folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ**, és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultsággal rendelkező folyamat automatikusan rendelkezik az összes alábbi jogosultsággal, a **PV_SU_** jogosultságok kivételével.

PV_AU_
Egyenértékű az összes **PV_AU_** jogosultság egyesítésével

PV_AU_ADD
Engedélyezi a folyamatnak a megfigyelési rekordok rögzítését/felvételét

PV_AU_ADMIN
Engedélyezi a folyamatnak a megfigyelési rendszer konfigurálását és lekérdezését

PV_AU_PROC
Engedélyezi a folyamatnak egy folyamat megfigyelési állapotának lekérdezését és beállítását

PV_AU_READ
Engedélyezi a folyamatnak a megfigyelési fájlként megjelölt fájlok olvasását

PV_AU_WRITE
Engedélyezi a folyamatnak a megfigyelési fájlként megjelölt fájlok írását és törlését, illetve fájlok megfigyelési fájlkénti megjelölését

Felhatalmazási jogosultságok:

A Trusted AIX rendszeren az alábbi felhatalmazási jogosultságok állnak rendelkezésre. Az alábbiakban megtalálja az egyes jogosultságok összegezését, leírását, illetve használatát. Bizonyos jogosultságok hierarchiát alkotnak. Ebben az esetben az egyik jogosultság egy másik jogosultsághoz tartozó minden jogot adományozhat.

A jogosultságok ellenőrzése során a rendszer először meghatározza, hogy a folyamat rendelkezik-e a szükséges legalacsonyabb jogosultsággal, majd megy felfele a hierarchiában az erősebb jogosultságok meglétének ellenőrzése érdekében. A **PV_AU_** jogosultsággal rendelkező folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ**, és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultsággal rendelkező folyamat automatikusan rendelkezik az összes alábbi jogosultsággal, a **PV_SU_** jogosultságok kivételével.

PV_AZ_ADMIN
Engedélyezi a folyamatnak a kernel biztonsági táblák módosítását

PV_AZ_READ
Engedélyezi a folyamatnak a kernel biztonsági táblák lekérését

PV_AZ_ROOT
A kapcsoló beállításának következtében a folyamatnak az **exec** rendszerhívás alatt felhatalmazási ellenőrzéseken kell átesnie

PV_AZ_CHECK
Engedélyezi a folyamatnak, hogy az összes felhatalmazási ellenőrzésen átmenjen

DAC jogosultságok:

A Trusted AIX rendszeren az alábbi DAC jogosultságok állnak rendelkezésre. Az alábbiakban megtalálja az egyes jogosultságok összegezését, leírását, illetve használatát. Bizonyos jogosultságok hierarchiát alkotnak. Ebben az esetben az egyik jogosultság egy másik jogosultsághoz tartozó minden jogot adományozhat.

A jogosultságok ellenőrzése során a rendszer először meghatározza, hogy a folyamat rendelkezik-e a szükséges legalacsonyabb jogosultsággal, majd megy felfele a hierarchiában az erősebb jogosultságok meglétének ellenőrzése érdekében. A **PV_AU_** jogosultsággal rendelkező folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ**, és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultsággal rendelkező folyamat automatikusan rendelkezik az összes alábbi jogosultsággal, a **PV_SU_** jogosultságok kivételével.

PV_DAC_
Egyenértékű az összes PV_DAC_ jogosultság egyesítésével

PV_DAC_O
Engedélyezi a folyamatnak a DAC tulajdonjog-korlátozások felülbírálását

PV_DAC_R
Engedélyezi a folyamatnak a DAC olvasással kapcsolatos korlátozások felülbírálását

PV_DAC_W
Engedélyezi a folyamatnak a DAC írással kapcsolatos korlátozások felülbírálását

PV_DAC_X
Engedélyezi a folyamatnak a DAC végrehajtással kapcsolatos korlátozások felülbírálását

PV_DAC_UID
Engedélyezi a folyamatnak saját felhasználói azonosítójának (UID) beállítását és módosítását

PV_DAC_GID
Engedélyezi a folyamatnak saját csoportazonosítójának (GID) beállítását és módosítását

PV_DAC_RID
Engedélyezi a folyamatnak saját szerepazonosítójának (RID) beállítását és módosítását

Fájlrendszer-jogosultságok:

A Trusted AIX rendszeren az alábbi fájlrendszer-jogosultságok állnak rendelkezésre. Az alábbiakban megtalálja az egyes jogosultságok összegezését, leírását, illetve használatát. Bizonyos jogosultságok hierarchiát alkotnak. Ebben az esetben az egyik jogosultság egy másik jogosultsághoz tartozó minden jogot adományozhat.

A jogosultságok ellenőrzése során a rendszer először meghatározza, hogy a folyamat rendelkezik-e a szükséges legalacsonyabb jogosultsággal, majd megy felfele a hierarchiában az erősebb jogosultságok meglétének ellenőrzése érdekében. A **PV_AU_** jogosultsággal rendelkező folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ**, és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultsággal rendelkező folyamat automatikusan rendelkezik az összes alábbi jogosultsággal, a **PV_SU_** jogosultságok kivételével.

PV_FS_
Egyenértékű az összes **PV_FS_** jogosultság egyesítésével

PV_FS_MKNOD
Engedélyezi a folyamatnak az **mknod** rendszerhívás végrehajtását és ezáltal tetszőleges típusú fájl létrehozását

PV_FS_MOUNT
Engedélyezi a folyamatnak egy fájlrendszer felépítését és lebontását

PV_FS_CHOWN
Engedélyezi a folyamatnak egy fájl tulajdonjogának módosítását

PV_FS_QUOTA
Engedélyezi a folyamatnak a lemezkvótákhoz kapcsolódó információk kezelését

PV_FS_LINKDIR
Engedélyezi a folyamatnak könyvtárra mutató közvetlen hivatkozás létrehozását

PV_FS_RESIZE
Engedélyezi a folyamatnak, hogy egy fájlrendszeren kiterjesztés és összehúzás típusú műveleteket végezzen

PV_FS_CNTL
Engedélyezi a folyamatnak, hogy egy fájlrendszeren - a kiterjesztés és összehúzás típusú műveletek kivételével - különböző vezérlőműveleteket végezzen

PV_FS_CHROOT
Engedélyezi a folyamatnak a gyökérkönyvtárának módosítását

PV_FS_PDMODE
Engedélyezi a folyamatnak particionált típusú könyvtár létrehozását és beállítását

Folyamatjogosultságok:

A Trusted AIX rendszeren az alábbi folyamatjogosultságok állnak rendelkezésre. Az alábbiakban megtalálja az egyes jogosultságok összegezését, leírását, illetve használatát. Bizonyos jogosultságok hierarchiát alkotnak. Ebben az esetben az egyik jogosultság egy másik jogosultsághoz tartozó minden jogot adományozhat.

A jogosultságok ellenőrzése során a rendszer először meghatározza, hogy a folyamat rendelkezik-e a szükséges legalacsonyabb jogosultsággal, majd megy felfele a hierarchiában az erősebb jogosultságok meglétének ellenőrzése érdekében. A **PV_AU_** jogosultsággal rendelkező folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ**, és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultsággal rendelkező folyamat automatikusan rendelkezik az összes alábbi jogosultsággal, a **PV_SU_** jogosultságok kivételével.

PV_PROC_
: Egyenértékű az összes **PV_PROC_** jogosultság egyesítésével

PV_PROC_PRIO
: Engedélyezi a folyamatnak vagy szálnak a prioritás, irányelv és más ütemezési paraméterek módosítását

PV_PROC_CORE
: Engedélyezi a folyamatnak a tárkiíratást

PV_PROC_RAC
: Engedélyezi a folyamatnak a felhasználónkénti korlátnál több folyamat létrehozását

PV_PROC_RSET
: Engedélyezi a folyamatnak erőforráshalmaz (**rset**) csatlakoztatását folyamathoz vagy szálhoz

PV_PROC_ENV
: Engedélyezi a folyamatnak felhasználói információk beállítását a felhasználói adatszerkezetben

PV_PROC_CKPT
: Engedélyezi a folyamatnak ellenőrzési pont készítését másik folyamatról vagy annak újraindítását

PV_PROC_CRED
: Engedélyezi a folyamatnak a folyamat hitelesítési attribútumainak beállítását

PV_PROC_SIG
: Lehetővé teszi a folyamat számára jelzés küldését egy nem kapcsolódó folyamatnak

PV_PROC_PRIV
: Engedélyezi a folyamatnak egy folyamathoz tartozó jogosultsághalmazok módosítását vagy megjelenítését

PV_PROC_TIMER
: Engedélyezi a folyamatnak finomabb beállítású időmérők elküldését és használatát

PV_PROC_RTCLK
: Engedélyezi a folyamatnak a CPU-idő óra elérését

PV_PROC_VARS
: Engedélyezi a folyamatnak a folyamat által hangolható paraméterek lekérését és frissítését

PV_PROC_PDMODE
: Engedélyezi a folyamatnak a particionált könyvtár VALÓS módjának módosítását

Kerneljogosultságok:

A Trusted AIX rendszeren az alábbi kerneljogosultságok állnak rendelkezésre. Az alábbiakban megtalálja az egyes jogosultságok összegezését, leírását, illetve használatát. Bizonyos jogosultságok hierarchiát alkotnak. Ebben az esetben az egyik jogosultság egy másik jogosultsághoz tartozó minden jogot adományozhat.

A jogosultságok ellenőrzése során a rendszer először meghatározza, hogy a folyamat rendelkezik-e a szükséges legalacsonyabb jogosultsággal, majd megy felfele a hierarchiában az erősebb jogosultságok meglétének ellenőrzése

érdekében. A **PV_AU_** jogosultsággal rendelkező folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ**, és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultsággal rendelkező folyamat automatikusan rendelkezik az összes alábbi jogosultsággal, a **PV_SU_** jogosultságok kivételével.

PV_KER_
Egyenértékű az összes **PV_KER_** jogosultság egyesítésével

PV_KER_ACCT
Engedélyezi a folyamatnak a számlázási alrendszerhez kapcsolódó korlátozott műveletek elvégzését

PV_KER_DR
Engedélyezi a folyamatnak a dinamikus újrakonfigurálási műveletek meghívását

PV_KER_TIME
Engedélyezi a folyamatnak a rendszerórát és -időt módosítását

PV_KER_RAC
Engedélyezi a folyamatnak nagy (nem lapozható) oldalak használatát az osztott memóriaszegmensekhez

PV_KER_WLM
Engedélyezi a folyamatnak a WLM konfiguráció inicializálását és módosítását

PV_KER_EWLM
Engedélyezi a folyamatnak az eWLM környezet inicializálását és lekérdezését

PV_KER_VARS
Engedélyezi a folyamatnak a kernel futási környezet beállítható paramétereinek megvizsgálását és beállítását

PV_KER_REBOOT
Engedélyezi a folyamatnak a rendszer leállítását

PV_KER_RAS
Engedélyezi a folyamatnak a RAS rekordok, a hibanaplózási, nyomkövetési és kiíratási funkciók beállítását és írását

PV_KER_LVM
Engedélyezi a folyamatnak az LVM alrendszer beállítását

PV_KER_NFS
Engedélyezi a folyamatnak az NFS alrendszer beállítását

PV_KER_VMM
Engedélyezi a folyamatnak a lapozási terület paramétereinek és a kernel egyéb VMM beállítható paramétereinek módosítását

PV_KER_WPAR
Engedélyezi a folyamatnak munkaterület-partíció beállítását

PV_KER_CONF
Engedélyezi a folyamatnak különböző rendszerkonfigurációs műveletek végrehajtását

PV_KER_EXTCONF
Engedélyezi egy folyamatnak, hogy a kernelbővítményekben különböző konfigurációs feladatokat végezzen

PV_KER_IPC
Engedélyezi egy folyamatnak, hogy az IPC üzenetsorpuffer értékét növelje, illetve engedélyezi az **shmget** rendszerhívásokat a csatolt tartományokkal

PV_KER_IPC_R
Engedélyezi a folyamatnak az IPC üzenetsor, szemaforhalmaz vagy osztott memóriaszegmens olvasását

PV_KER_IPC_W
Engedélyezi a folyamatnak az IPC üzenetsor, szemaforhalmaz vagy osztott memóriaszegmens írását

PV_KER_IPC_O
Engedélyezi a folyamatnak, hogy az összes IPC objektum DAC tulajdonjogát felülbírálja

PV_KER_SECCONFIG
Engedélyezi a folyamatnak a kernel biztonsági kapcsolóinak beállítását

PV_KER_PATCH
Engedélyezi a folyamatnak a kernelbővítmények javítását

Címke jogosultságok:

A Trusted AIX rendszeren az alábbi címke jogosultságok állnak rendelkezésre. Az alábbiakban megtalálja az egyes jogosultságok összegezését, leírását, illetve használatát. Bizonyos jogosultságok hierarchiát alkotnak. Ebben az esetben az egyik jogosultság egy másik jogosultsághoz tartozó minden jogot adományozhat.

A jogosultságok ellenőrzése során a rendszer először meghatározza, hogy a folyamat rendelkezik-e a szükséges legalacsonyabb jogosultsággal, majd megy felfele a hierarchiában az erősebb jogosultságok meglétének ellenőrzése érdekében. A **PV_AU_** jogosultsággal rendelkező folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ**, és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultsággal rendelkező folyamat automatikusan rendelkezik az összes alábbi jogosultsággal, a **PV_SU_** jogosultságok kivételével.

PV_LAB_
Egyenértékű az összes egyéb címke jogosultság (**PV_LAB_***) egyesítésével

PV_LAB_CL
Engedélyezi a folyamatnak az alany SCL-ek módosítását, a folyamat engedélyének függvényében

PV_LAB_CLTL
Engedélyezi a folyamatnak az alany TCL-ek módosítását, a folyamat engedélyének függvényében

PV_LAB_LEF
Engedélyezi a folyamatnak a címkekezelési adatbázis olvasását

PV_LAB_SLDG
Engedélyezi a folyamatnak az SL-ek visszaléptetését a folyamat engedélyének függvényében

PV_LAB_SLDG_STR
Engedélyezi a folyamatnak egy csomag SL-jének visszaléptetését a folyamat engedélyének függvényében

PV_LAB_SL_FILE
Engedélyezi a folyamatnak az objektum SL-ek módosítását a folyamat engedélyének függvényében

PV_LAB_SL_PROC
Engedélyezi a folyamatnak az alany SL-ek módosítását a folyamat engedélyének függvényében

PV_LAB_SL_SELF
Engedélyezi a folyamatnak a saját SL-jének módosítását a folyamat engedélyének függvényében

PV_LAB_SLUG
Engedélyezi a folyamatnak az SL-ek kiemelését a folyamat engedélyének függvényében

PV_LAB_SLUG_STR
Engedélyezi a folyamatnak egy csomag SL-jének kiemelését a folyamat engedélyének függvényében

PV_LAB_TL
Engedélyezi a folyamatnak az alany és objektum TL módosítását

MAC jogosultságok:

A Trusted AIX rendszeren az alábbi MAC jogosultságok állnak rendelkezésre. Az alábbiakban megtalálja az egyes jogosultságok összegezését, leírását, illetve használatát. Bizonyos jogosultságok hierarchiát alkotnak. Ebben az esetben az egyik jogosultság egy másik jogosultsághoz tartozó minden jogot adományozhat.

A jogosultságok ellenőrzése során a rendszer először meghatározza, hogy a folyamat rendelkezik-e a szükséges legalacsonyabb jogosultsággal, majd megy felfele a hierarchiában az erősebb jogosultságok meglétének ellenőrzése érdekében. A **PV_AU_** jogosultsággal rendelkező folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ**, és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultsággal rendelkező folyamat automatikusan rendelkezik az összes alábbi jogosultsággal, a **PV_SU_** jogosultságok kivételével.

PV_MAC_
: Egyenértékű az összes egyéb MAC jogosultság (**PV_MAC_***) egyesítésével

PV_MAC_CL
: Engedélyezi a folyamatnak az érzékenységengedély-korlátozások kihagyását

PV_MAC_R_PROC
: Engedélyezi a folyamatnak a MAC olvasási megszorítások kihagyását folyamatinformációk lekérésekor, ha a célfolyamat címkéje a működő folyamat engedélyén belül van

PV_MAC_W_PROC
: Engedélyezi a folyamatnak a MAC írási megszorítások kihagyását jelzés küldésekor a folyamatnak, ha a célfolyamat címkéje a működő folyamat engedélyén belül van

PV_MAC_R
: Engedélyezi a folyamatnak a MAC olvasási korlátozások kihagyását

PV_MAC_R_CL
: Engedélyezi a folyamatnak a MAC olvasási megszorítások kihagyását, ha az objektum címkéje a folyamat engedélyén belül van

PV_MAC_R_STR
: Engedélyezi a folyamatnak a MAC olvasási megszorítások kihagyását üzenet olvasásakor az adatfolyamból, ha az üzenet címkéje a folyamat engedélyén belül van

PV_MAC_W
: Engedélyezi a folyamatnak a MAC írási korlátozások kihagyását

PV_MAC_W_CL
: Engedélyezi a folyamatnak a MAC írási megszorítások kihagyását, ha az objektum címkéje a folyamat engedélyén belül van

PV_MAC_W_DN
: Engedélyezi a folyamatnak a MAC írási megszorítások kihagyását, ha a folyamat címkéje dominál az objektum címkéjével szemben és az objektum címkéje a folyamat engedélyén belül van

PV_MAC_W_UP
: Engedélyezi a folyamatnak a MAC írási megszorítások kihagyását, ha a folyamat címkéjével szemben az objektum címkéje dominál és az objektum címkéje a folyamat engedélyén belül van

PV_MAC_OVRRD
: Kihagyja a MAC korlátozásokat a MAC alól kivételként jelzett fájlok esetén

MIC jogosultságok:

A Trusted AIX rendszeren az alábbi MIC jogosultságok állnak rendelkezésre. Az alábbiakban megtalálja az egyes jogosultságok összegezését, leírását, illetve használatát. Bizonyos jogosultságok hierarchiát alkotnak. Ebben az esetben az egyik jogosultság egy másik jogosultsághoz tartozó minden jogot adományozhat.

A jogosultságok ellenőrzése során a rendszer először meghatározza, hogy a folyamat rendelkezik-e a szükséges legalacsonyabb jogosultsággal, majd megy felfele a hierarchiában az erősebb jogosultságok meglétének ellenőrzése érdekében. A **PV_AU_** jogosultsággal rendelkező folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ**, és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultsággal rendelkező folyamat automatikusan rendelkezik az összes alábbi jogosultsággal, a **PV_SU_** jogosultságok kivételével.

PV_MIC
Engedélyezi a folyamatnak az integritási korlátozások kihagyását

PV_MIC_CL
Engedélyezi a folyamatnak az integritási engedélykorlátozások kihagyását

Hálózati jogosultságok:

A Trusted AIX rendszeren az alábbi hálózati jogosultságok állnak rendelkezésre. Az alábbiakban megtalálja az egyes jogosultságok összegezését, leírását, illetve használatát. Bizonyos jogosultságok hierarchiát alkotnak. Ebben az esetben az egyik jogosultság egy másik jogosultsághoz tartozó minden jogot adományozhat.

A jogosultságok ellenőrzése során a rendszer először meghatározza, hogy a folyamat rendelkezik-e a szükséges legalacsonyabb jogosultsággal, majd megy felfele a hierarchiában az erősebb jogosultságok meglétének ellenőrzése érdekében. A **PV_AU_** jogosultsággal rendelkező folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ**, és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultsággal rendelkező folyamat automatikusan rendelkezik az összes alábbi jogosultsággal, a **PV_SU_** jogosultságok kivételével.

PV_NET_
Egyenértékű az összes egyéb hálózati jogosultság (**PV_NET_***) egyesítésével

PV_NET_CNTL
Engedélyezi a folyamatnak a hálózati táblázatok módosítását

PV_NET_PORT
Engedélyezi a folyamatnak a csatlakozást a korlátozott portokon

PV_NET_RAWSOCK
Engedélyezi a folyamatnak a hálózati réteg közvetlen elérését

PV_NET_CONFIG
Engedélyezi a folyamatnak a hálózati paraméterek beállítását

Felettes felhasználó jogosultságai:

A Trusted AIX rendszeren az alábbi felettes felhasználó jogosultságok állnak rendelkezésre. Az alábbiakban megtalálja az egyes jogosultságok összegezését, leírását, illetve használatát. Bizonyos jogosultságok hierarchiát alkotnak. Ebben az esetben az egyik jogosultság egy másik jogosultsághoz tartozó minden jogot adományozhat.

A jogosultságok ellenőrzése során a rendszer először meghatározza, hogy a folyamat rendelkezik-e a szükséges legalacsonyabb jogosultsággal, majd megy felfele a hierarchiában az erősebb jogosultságok meglétének ellenőrzése érdekében. A **PV_AU_** jogosultsággal rendelkező folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ**, és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultsággal rendelkező folyamat automatikusan rendelkezik az összes alábbi jogosultsággal, a **PV_SU_** jogosultságok kivételével.

PV_SU_
Egyenértékű az összes egyéb Felettes felhasználói jogosultság (**PV_MAC_***) egyesítésével

PV_SU_ROOT
A folyamatnak megadja a szabványos felettes felhasználóhoz tartozó összes jogosultságot

PV_SU_EMUL
A folyamatnak megadja a szabványos felettes felhasználóhoz tartozó összes jogosultságot, amikor a folyamat UID azonosítója 0

PV_SU_UID
Hatására a **getuid** rendszerhívás 0 értéket ad vissza

Egyéb jogosultságok:

A Trusted AIX rendszeren az alábbi egyéb jogosultságok állnak rendelkezésre. Az alábbiakban megtalálja az egyes jogosultságok összegezését, leírását, illetve használatát. Bizonyos jogosultságok hierarchiát alkotnak. Ebben az esetben az egyik jogosultság egy másik jogosultsághoz tartozó minden jogot adományozhat.

A jogosultságok ellenőrzése során a rendszer először meghatározza, hogy a folyamat rendelkezik-e a szükséges legalacsonyabb jogosultsággal, majd megy felfele a hierarchiában az erősebb jogosultságok meglétének ellenőrzése érdekében. A **PV_AU_** jogosultsággal rendelkező folyamat például automatikusan rendelkezik a **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ**, és **PV_AU_WRITE** jogosultsággal, illetve a **PV_ROOT** jogosultsággal rendelkező folyamat automatikusan rendelkezik az összes alábbi jogosultsággal, a **PV_SU_** jogosultságok kivételével.

PV_ROOT
: A folyamatnak megadja az összes jogosultságot, a **PV_SU_** (és a **PV_SU_** által meghatározott jogosultságok) kivételével

PV_TCB
: Engedélyezi a folyamatnak a kernel megbízható könyvtár-útvonalainak módosítását

PV_TP
: Jelzi, hogy a folyamat megbízható útvonalú folyamat és lehetővé teszi a megbízható útvonalú folyamatokra korlátozott műveleteket

PV_TP_SET
: Engedélyezi a folyamatnak a kernel megbízható útvonal kapcsoló beállítását vagy törlését

PV_WPAR_CKPT
: Engedélyezi a folyamatnak az ellenőrzési pontot és újraindítást érintő műveletek végrehajtását a munkaterület-partíción

PV_DEV_CONFIG
: Engedélyezi a folyamatnak a rendszer kernelbővítmények és -eszközök beállítását

PV_DEV_LOAD
: Engedélyezi a folyamatnak a rendszer kernelbővítmények és -eszközök betöltését és eltávolítását a rendszeren

PV_DEV_QUERY
: Engedélyezi a folyamatnak a kernelmodulok lekérdezését

Megbízható AIX hibáinak elhárítása

Az általános kérdések válaszai segítséget nyújthatnak a Trusted AIX hibaelhárításában.

Hogyan jelentkezhetek be a Trusted AIX rendszerbe?
: A Trusted AIX három adminisztrációs felhasználót hoz létre a telepítés során a lentiek szerint megadott megfelelő szerepekkel.

  Ezen fiókok jelszavát akkor kell beállítani, amikor először valósul meg rendszerbetöltés Trusted AIX telepítés után. Ha a rendszert a hálózatról beavatkozás nélküli módban telepítette, akkor ezen alapértelmezett fiókok jelszava alább látható.

Felhasználó	Jelszó
isso	isso
sa	sa
so	so

Hogyan lehet su paranccsal root felhasználóra átkapcsolni?
Trusted AIX telepítésekor a root **su** attribútuma **false**, így egyik felhasználó sem tudja elérni ezt a fiókot. A fiók eléréséhez az alapértelmezett adminisztrátori felhasználóknak, valamint az isso és sa felhasználónak a **chuser** paranccsal módosítania kell a root fiók attribútumát **true** értékre.

Ha a **su** a root számára engedélyezett és a root fiók jelszava nincs beállítva, akkor a rendszer minden felhasználója hozzá tud férni a root fiókhoz. Ennek elkerülése érdekében a **su** attribútum visszaállása előtt a root fiók jelszavát be kell állítani

Létre kell hoznom saját adminisztrátori felhasználókat vagy használjam az alapértelmezetteket?
Az alapértelmezett adminisztrátori felhasználók csak a rendszer személyre szabásához vannak beállítva. Ez nagyon ajánlott, de nem szükséges, hogy a fiókokat csak a rendszer személyre szabásához használja.

Hozzon létre saját adminisztrátori felhasználókat megfelelő isso, sa és so szereppel, és törölje vagy tiltsa le ezeket az alapértelmezett felhasználókat.

Miért nem tudok bejelentkezni a rendszerre?
Ha megpróbál rootként (uid 0 fiók) vagy 128-nál kisebb uid-vel rendelkező fiókkal bejelentkezni, akkor a hozzáférést a rendszer megtagadja. Ezeket a fiókokat rendszerfiókoknak hívjuk. A rendszerfiókok eléréséhez nem rendszerfiók-felhasználóként kell bejelentkezni és **su** parancsot kell átjelentkezni rá.

A bejelentkezés közben megjelenített címkekódolási fájl hibás?
Ha a címkekódolási fájl sérült, akkor be kell lépnie egyfelhasználós módba root felhasználóként. A root fiók csak egyfelhasználós módban érhető el.

A **labck** paranccsal ellenőrizze, hogy a címkekódolási fájl (/etc/security/enc/LabelEncodings) megfelelő-e. Ha a fájl helytelen, akkor módosítsa a fájlt és végezze el újra az ellenőrzést a **labck** paranccsal az egyfelhasználós módból való kilépés előtt.

Futtassa a **trustchk** parancsot interaktív módban (**trustchk -t ALL**) a rendszer állapotának ellenőrzése érdekében.

Trusted AIX rendszeren miért nem tudok Trusted AIX függvénytár API-kat használó programokat fordítani?
A fejlesztői eszközkészlet alapértelmezésben nem kerül telepítésre. A bos.mls.adt fájlkészletet a telepítési adathordozóról kell telepíteni.

Hogyan lehet kijavítani a parancsjogosultságok azon módosításait, amelyek a parancsok nem megfelelő működését okoztál?
Futtassa a **trustchk** parancsot interaktív módban (**trustchk -t**) ezekhez a parancsokhoz a jogosultságok kijavítása érdekében.

Az /etc/security/enc könyvtár miért nem elérhető?
Az /etc/security/enc könyvtár eléréséhez a parancsértelmező PV_LAB_LEF és PV_MAC_R jogosultságot igényel. Rendelje hozzá ezeket a jogosultságokat a parancsértelmezőhöz.

Hogyan tiltható le a trustchk rendszerbetöltéskor?
Távolítsa el vagy tegye megjegyzésbe a trustchk sort az /etc/rc.mls parancsfájlban.

Hogyan akadályozható meg, hogy a rendszer rendszerbetöltési hitelesítést kérjen minden rendszerbetöltéskor?
Elképzelhető, hogy engedélyezte a rendszerbetöltési hitelesítést a rendszerhez. A Trusted AIX almenü SMIT menüjében letilthatja.

Miért nem működik a módosítás a fájlrendszer-objektum SL-jének módosítására tett kísérlet esetén?
Számos lehetőség van:

Az /usr/sbin/settxattr hibaüzeneteket ad vissza?
Ha igen, akkor további információkért tekintse meg azokat. Például:

Jogosult az /usr/sbin/settxattr végrehajtására?
Ha nem, akkor ellenőrizze a jogosultságokat és hitelesítéseket.

A szintaxis helyes?
A szintaxist a **settxattr** man oldalán tekintheti meg.

A kért SL vagy annak rövidítése létezik?
A "con a b" kérése a rendszeren alapértelmezett címkekódolási fájllal működik (/etc/security/enc/LabelEncodings), de a "conf a b" kérése nem, még akkor sem, ha mindkettő a "confidential compartment A compartment B" logikai rövidítésének tűnik.

Többszavas címkéhez kell idézőjelet használni?
A settxattr -f sl=con <*fájlnév*> működik, a settxattr -f -a sl="con a b" <*fájlnév*> szintén, de a settxattr -a sl=con a b <*fájlnév*> nem.

A settxattr visszaad hibaüzeneteket?
Ha nem adott vissza hibaüzeneteket, akkor elképzelhető, hogy a fájlrendszer-objektum szimbolikus hivatkozás. Ha a módosítani próbált objektum szimbolikus hivatkozás, akkor először határozza meg, hogy a hivatkozásnak az SL-jét kívánja módosítani vagy magát az objektumot, amelyre a hivatkozás mutat. A **settxattr** nem követi a hivatkozásokat, hanem beállítja a hivatkozás címkéit.

Hogyan telepíthető egy harmadik féltől származó alkalmazás, hogy megfelelően működjön a rendszeren?
Ha telepített egy harmadik féltől származó alkalmazást és az nem működik megfelelően, akkor elképzelhető, hogy az olyan korlátozott fájlokat vagy könyvtárakat ér el, amelyek extra jogosultságokat igényelnek. Az alkalmazás korlátozott objektumok elérése iránti igényeinek kiértékelése után határozza meg a szükséges jogosultságokat az alábbi módon.

- Rendeljen PV_ROOT attribútumot a parancsértelmezőhöz
- Futtassa a tracepriv -f -e <*harmadik féltől származó parancs*> parancssort

Ez listázza az alkalmazás által igényelt jogosultságot. Adja hozzá ezeket a privilegizált parancsadatbázishoz a **setsecattr** paranccsal.

Miért nem tudok bizonyos parancsokat végrehajtani?
Mivel a legtöbb parancsot felhatalmazások védik, a privilegizált parancsok egy részének végrehajtása csak akkor engedélyezett, ha a meghívó felhasználó rendelkezik a megfelelő felhatalmazással. Ehhez ellenőrizze, hogy a parancs végrehajtásához szükséges felhatalmazás létezik-e az aktuális munkamenethez aktivált szerepek egyikében.

Ellenőrizze az aktív felhatalmazásokat a rolelist -ae paranccsal és a parancs által kívánt felhatalmazást az lssecattr -c <*parancs*> paranccsal.

Néhány parancs miért nem jeleníti meg megfelelően a címkéket?
Ezen parancsok nagy része az /etc/security/enc/LabelEncodings fájlt használja a címkék felhasználó által olvasható formátumra alakításához és visszaalakításához. Ha a fájl sérült vagy módosítva lett, akkor elképzelhető, hogy a parancsok nem a várt módon működnek.

Fájlbiztonsági kapcsolók

A fájlbiztonsági kapcsolók befolyásolják a fájl elérésének módját. A kapcsolók a fájl kiterjesztett attribútumainak (EA) részeként kerülnek tárolásra. A fájlbiztonsági kapcsolók meghatározását a fejlécfájl tartalmazza.

FSF_APPEND
Működési módban a fájlhoz csak hozzáfűzni lehet, a fájl nem változtatható meg.

FSF_AUDIT
A fájl a megfigyelési alrendszer részeként került megjelölésre. Az ilyen fájlok írásához és olvasásához a folyamatnak **PV_AU_READ**, illetve **PV_AU_WRITE** jogosultsággal kell rendelkeznie, értelemszerűen.

FSF_MAC_EXMPT
A **PV_MAC_OVRRD** jogosultsággal rendelkező ESP figyelmen kívül hagyja a MAC megszorítást az objektum elérésére tett kísérlet során.

FSF_PDIR
A könyvtár particionált könyvtár.

FSF_PSDIR
A könyvtár particionált alkönyvtár.

FSF_PSSDIR
A könyvtár particionált al-alkönyvtár.

FSF_TLIB
Az objektum a megbízható könyvtár részeként került megjelölésre. A számítógépnek konfigurációs módban kell futnia, vagy a **trustedlib_enabled** kernelbiztonsági kapcsoló állapota OFF kell, hogy legyen.

FSF_TLIB_PROC
A TLIB folyamatként megjelölt folyamatok csak az olyan ***.so** könyvtárakhoz csatolhatók, amelyek **TLIB** kapcsolója beállított. A rendszernek konfigurációs módban kell futnia, vagy a **trustedlib_enabled** kernelbiztonsági kapcsoló állapota OFF kell, hogy legyen.

Trusted AIX parancsok

Biztonsággal kapcsolatos parancsok biztosítottak a Trusted AIX rendszer kezeléséhez:

labck Ellenőrzi a LabelEncodings fájlt

getsecconf
Megjeleníti a kernelbiztonsági jelzőket

setsecconf
Módosítja a Trusted AIX kernel biztonsági jelzőit

getsyslab
Megjeleníti a kernel maximális és minimális címkéit

setsyslab
Beállítja a kernel maximális és minimális címkéit

getrunmode
Megjeleníti a rendszer aktuális futási módját

setrunmode
Átkapcsolja a rendszer futási módját

pdlink Csatolja a fájlokat különböző particionált alkönyvtárakon keresztül

pdmkdir
Particionált könyvtárakat és alkönyvtárakat hoz létre

pdmode
Visszaadja az aktuális particionált könyvtárhozzáférési módot vagy egy parancsot megadott particionált könyvtárhozzáférési mód nélkül futtat

pdrmdir
Particionált könyvtárakat és társított alkönyvtárakat távolít el

pdset Particionált (al)könyvtárakat állít be, vagy megszünteti azok beállítását

bootauth
Ellenőrzi, hogy felhatalmazott felhasználó tölti-e be a rendszert

chuser Módosítja a felhasználó engedély attribútumait

lsuser Megjeleníti a felhasználó engedély attribútumait

chsec Módosítja a felhasználó engedély attribútumait és portcímkéit

lssec Megjeleníti a felhasználó engedély attribútumait és portcímkéit

trustchk
Ellenőrzi a fájlok attribútumait

lstxattr
Megjeleníti a fájlok, folyamatok és IPC objektumok címkéjét és biztonsági jelző attribútumait

settxattr
Módosítja a fájlok, folyamatok és IPC objektumok címkéjét és biztonsági jelző attribútumait

Nyilatkozatok

Ezek az információk az Egyesült Államokban forgalmazott termékekre és szolgáltatásokra vonatkoznak.

Elképzelhető, hogy a dokumentumban tárgyalt termékeket, szolgáltatásokat vagy lehetőségeket az IBM más országokban nem forgalmazza. Az adott országokban rendelkezésre álló termékekről és szolgáltatásokról az IBM helyi képviseletei szolgálnak felvilágosítással. Az IBM termékeire, programjaira vagy szolgáltatásaira vonatkozó utalások sem állítani, sem sugallni nem kívánják, hogy az adott helyzetben csak az adott IBM termék, program vagy szolgáltatás alkalmazható. Minden olyan működésében azonos termék, program vagy szolgáltatás alkalmazható, amely nem sérti az IBM szellemi tulajdonjogát. A nem IBM termékek, programok és szolgáltatások működésének megítélése és ellenőrzése azonban a felhasználó felelőssége.

A dokumentum tartalmával kapcsolatban az IBM bejegyzett vagy bejegyzés alatt álló szabadalmakkal rendelkezhet. Jelen dokumentum nem ad semmiféle jogos licencet ezen szabadalmakhoz. A licenckérelmeket írásban a következő címre küldheti:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Ha duplabyte-os (DBCS) karakterkészlettel kapcsolatban van szüksége licencre, akkor lépjen kapcsolatba az országában az IBM szellemi tulajdon osztályával, vagy írjon a következő címre:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

AZ IBM A KIADVÁNYT "JELENLEGI FORMÁJÁBAN", BÁRMIFÉLE KIFEJEZETT VAGY VÉLELMEZETT GARANCIA NÉLKÜL ADJA KÖZRE, IDEÉRTVE, DE NEM KIZÁRÓLAG A JOGSÉRTÉS KIZÁRÁSÁRA, A KERESKEDELMI ÉRTÉKESÍTHETŐSÉGRE ÉS BIZONYOS CÉLRA VALÓ ALKALMASSÁGRA VONATKOZÓ VÉLELMEZETT GARANCIÁT. Bizonyos államok nem engedélyezik egyes tranzakciók kifejezett vagy vélelmezett garanciáinak kizárását, így elképzelhető, hogy az előző bekezdés Önre nem vonatkozik.

Jelen dokumentum tartalmazhat technikai pontatlanságokat és sajtóhibákat. A kiadványban leírt információk bizonyos időnként módosításra kerülnek; a módosításokat a kiadvány új kiadásai tartalmazzák. Az IBM mindennemű értesítés nélkül fejlesztheti és/vagy módosíthatja a kiadványban tárgyalt termékeket és/vagy programokat.

A kiadványban a nem az IBM által üzemeltetett webhelyek megjelenése csak kényelmi célokat szolgál, és semmilyen módon nem jelenti e webhelyek előnyben részesítését másokhoz képest. Az ilyen webhelyeken található anyagok nem képezik az adott IBM termék dokumentációjának részét, így ezek felhasználása csak saját felelősségre történhet.

Az IBM belátása szerint bármilyen formában felhasználhatja és továbbadhatja a felhasználóktól származó információkat anélkül, hogy a felhasználó felé ebből bármilyen kötelezettsége származna.

A programlicenc azon birtokosai, akik (i) a függetlenül létrehozott programok vagy más programok (beleértve ezt a programot is) közti információcsere, illetve (ii) a kicserélt információk kölcsönös használata céljából szeretnének információkhoz jutni, a következő címre írjanak:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Az ilyen információk bizonyos feltételek és kikötések mellett állnak rendelkezésre, ideértve azokat az eseteket is, amikor ez díjfizetéssel jár.

Az IBM a dokumentumban tárgyalt licencprogramokat és a hozzájuk tartozó licenc anyagokat IBM Vásárlói megállapodás, IBM Nemzetközi programlicenc szerződés vagy a felek azonos tartalmú megállapodása alapján biztosítja.

Az említett teljesítményadatok és ügyfélpéldák csak szemléltetési célt szolgálnak. A tényleges teljesítményadatok az adott konfigurációtól és működési feltételektől függően változhatnak.

A nem IBM termékekre vonatkozó információk a termékek szállítóitól, illetve azok publikált dokumentációiból, valamint egyéb nyilvánosan hozzáférhető forrásokból származnak. Az IBM nem tesztelte ezeket a termékeket, így a más gyártótól származó termékek esetében nem tudja megerősíteni a teljesítményre és kompatibilitásra vonatkozó, valamint az egyéb állítások pontosságát. A nem IBM termékekkel kapcsolatos kérdéseivel forduljon az adott termék szállítóihoz.

Az IBM jövőbeli tevékenységére vagy szándékaira vonatkozó állításokat az IBM mindennemű értesítés nélkül módosíthatja, azok csak célokat jelentenek.

A közölt IBM árak az IBM javasolt kiskereskedelmi árai, amelyek előzetes értesítés nélkül megváltozhatnak. Az egyes viszonteladók árai eltérhetnek ettől.

A leírtak csak tervezési célokat szolgálnak. Az információk a tárgyalt termékek elérhetővé válása előtt megváltozhatnak.

Az információk között példaként napi üzleti tevékenységekhez kapcsolódó jelentések és adatok lehetnek. A valóságot a lehető legjobban megközelítő illusztráláshoz a példákban egyének, vállalatok, márkák és termékek nevei szerepelnek. Minden ilyen név a képzelet szüleménye, és az esetleges hasonlóságuk a valódi személyekhez és üzleti vállalkozásokhoz teljes egészében a véletlen műve.

Szerzői jogi licenc:

A kiadvány forrásnyelvi alkalmazásokat tartalmaz, amelyek a programozási technikák bemutatására szolgálnak a különböző működési környezetekben. A példaprogramokat tetszőleges formában, az IBM -nek való díjfizetés nélkül másolhatja, módosíthatja és terjesztheti fejlesztés, használat, eladás vagy a példaprogramot futtató operációs rendszer alkalmazásprogramozási felületének megfelelő alkalmazásprogram terjesztésének céljából. Ezek a példák nem kerültek minden körülmények között tesztelésre. Ennek megfelelően az IBM nem tudja garantálni a programok megbízhatóságát, használhatóságát és működését. A példaprogramok "JELENLEGI FORMÁJUKBAN", bármiféle garancia nélkül kerülnek közreadásra. Az IBM semmilyen felelősséggel nem tartozik a példaprogramok használatából adódó esetleges károkért.

A példaprogramok minden másolatának, bármely részletének, illetve az ezek felhasználásával készült minden származtatott munkának tartalmaznia kell az alábbi szerzői jogi feljegyzést:

© (cégnév) (évszám).

A kód bizonyos részei az IBM Corp. példaprogramjaiból származnak.

© Copyright IBM Corp. (évszám vagy évszámok).

Adatvédelmi irányelv szempontok

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as the customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Védjegyek

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

A Microsoft és a Windows a Microsoft Corporation védjegye az Egyesült Államokban és/vagy más országokban.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Tárgymutató

Különleges jelek

A, Á

B

C

CS

D

E, É

F

H

I, Í

J

K

L

M

N

NY

O, Ó

P

R

S

SZ

T

V

W

X

Nyomtatva Dániában