AIX, verzia 7.2

Bezpečnosť

AIX, verzia 7.2

Bezpečnosť

Poznámka

Pred použitím týchto informácií a produktu, na ktorý sa vzťahujú, si prečítajte informácie v časti “Vyhlásenia” na strane 487.

Toto vydanie sa vzťahuje na produkt AIX verzie 7.2 a na všetky následné vydania a modifikácie, pokiaľ v nových vydaniach nebude uvedené inak.

Obsah

O tomto dokumente

Táto kolekcia tém poskytuje administrátorom systémov podrobné informácie o zabezpečení súborov, systému a siete. Obsahuje informácie o tom, ako môžete vykonávať rôzne úlohy, ako sú posilnenie zabezpečenia systému, zmena oprávnení, konfigurácia autentifikačných metód a konfigurácia súčastí Common Criteria Security Evaluation. Táto kolekcia tém je k dispozícii aj na disku CD s dokumentáciou, ktorý bol dodaný s operačným systémom.

Zvýraznenia textu

V tomto dokumente sa používajú nasledujúce spôsoby zvýrazňovania:

Tučné písmo	Identifikuje príkazy, podrutiny, kľúčové slová, súbory, štruktúry, adresáre a iné položky, ktorých názvy sú preddefinované systémom. Tiež identifikuje grafické objekty, ako sú tlačidlá, návestia a ikony, ktoré vyberá používateľ.
Kurzíva	Identifikuje parametre, ktorých skutočné názvy alebo hodnoty určuje používateľ.
`Monospace`	Označuje príklady špecifických hodnôt údajov, príklady textu podobného tomu, ktorý sa môže zobraziť, príklady častí kódu programu podobné tomu, ktorý môže napísať programátor, správy zo systému alebo informácie, ktoré by ste v skutočnosti mali napísať.

Rozlišovanie veľkosti písmen v systéme AIX

V rámci celého operačného systému AIX sa rozlišujú malé a veľké písmená. Napríklad, pomocou príkazu **ls** môžete zobraziť zoznam súborov. Ak zadáte príkaz LS, systém zobrazí správu, že príkaz sa nenašiel. Podobne **FILEA**, **FiLea** a **filea** sú názvy troch rôznych súborov, a to aj v prípade, že sa nachádzajú v tom istom adresári. Vždy sa preto uistite, že používate správnu veľkosť písmen, aby ste predišli vykonaniu neželaných úkonov.

ISO 9000

Pri vývoji a výrobe tohto produktu boli použité systémy s certifikáciou kvality ISO 9000.

Bezpečnosť

Operačný systém AIX vám umožňuje vykonávať rôzne úlohy, ako napríklad posilnenie zabezpečenia systému, zmena oprávnení, konfigurácia autentifikačných metód a konfigurácia súčastí Common Criteria Security Evaluation. Táto kolekcia tém je k dispozícii aj na disku CD s dokumentáciou, ktorý bol dodaný s operačným systémom.

Súvisiace informácie:

- Computer Emergency Response Team na univerzite Carnegie Mellon University (CERT)
- Forum of Incident Response and Security Teams (FIRST)
- Center for Education and Research in Information Assurance and Security (CERIAS)

Čo je nové v oblasti bezpečnosti

Prečítajte si nové alebo vo veľkej miere zmenené informácie v kolekcii tém Bezpečnosť.

Ako zistíte, čo je nové a čo sa zmenilo

V tomto súbore PDF môžete vidieť čiary revízií (|) na ľavom okraji, ktoré označujú nové alebo zmenené informácie.

Január 2017

Nasledujúce informácie predstavujú zhrnutie zmien v tejto kolekcii tém:

- Boli pridané informácie o udalostiach auditu v téme “Udalosti auditu” na strane 134.
- Boli pridané informácie o obrazoch softvéru OpenSSH v téme “Obrazy OpenSSH” na strane 193.

Zabezpečenie základného operačného systému

Téma Zabezpečenie základného operačného systému poskytuje informácie o tom, ako môžete chrániť svoj systém bez ohľadu na spôsob pripojenia k sieti.

Táto časť popisuje, ako inštalovať váš systém so zapnutými bezpečnostnými voľbami a ako zabezpečiť AIX, aby neprivilegovaní užívatelia nezískali prístup do vášho systému.

Bezpečná inštalácia a konfigurácia systému

Bezpečnosť inštalácie a konfigurácie AIX ovplyvňuje niekoľko faktorov.

Trusted Computing Base

Systémový administrátor musí určiť, aký stupeň dôveryhodnosti je možné priradiť konkrétnemu programu. Pri určovaní stupňa dôveryhodnosti požadovaného pre program, ktorý sa má inštalovať s oprávnením, treba zohľadniť hodnotu informačných prostriedkov v systéme.

TCB (Trusted Computing Base) je súčasťou systému, ktorý zodpovedá za vynucovanie celosystémových politík pre bezpečnosť informácií. Inštaláciou a použitím TCB môžete definovať prístup užívateľa k dôveryhodnej komunikačnej ceste, ktorá povoľuje bezpečnú komunikáciu medzi užívateľmi a TCB. Funkcie súčasti TCB je možné povoliť len v prípade, že je nainštalovaný operačný systém. Pri inštalácii súčasti TCB na už nainštalovaný počítač bude nutné vykonať inštaláciu typu Preservation. Zapnutie TCB vám povolí prístup do dôveryhodného prostredia shell, k dôveryhodným procesom a k SAK (Secure Attention Key).

Kontrola súčasti TCB:

Bezpečnosť operačného systému je ohrozená, keď súbory TCB (Trusted Computing Base) nie sú správne chránené, alebo keď konfiguračné súbory nemajú bezpečné hodnoty.

Príkaz **tcbck** vykonáva audit stavu zabezpečenia súčasti Trusted Computing Base. Príkaz **tcbck** vykonáva audit týchto informácií čítaním súboru /etc/security/sysck.cfg. Tento súbor obsahuje popis všetkých súborov súčasti TCB, konfiguračných súborov a dôveryhodných príkazov.

Súbor /etc/security/sysck.cfg nie je offline, a preto existuje riziko, že ho pozmení hacker. Nezabudnite po každej aktualizácii TCB vytvoriť jeho offline kópiu určenú iba na čítanie. Taktiež nezabudnite pred vykonaním ľubovoľných kontrol skopírovať tento súbor z archivačného média na disk.

Štruktúra súboru sysck.cfg:

Príkaz **tcbck** prečíta súbor /etc/security/sysck.cfg, aby stanovil, ktoré súbory sa majú skontrolovať. Každý dôveryhodný program v systéme je popísaný sekciou v súbore /etc/security/sysck.cfg.

Každá sekcia má nasledovné atribúty:

Atribút	Description
acl	Textový reťazec reprezentujúci zoznam prístupových práv (ACL - access control list) pre súbor. Musí byť v rovnakom formáte ako výstup príkazu **aclget**. Ak sa nezhoduje so skutočným súborom ACL (zoznam prístupových práv), príkaz **sysck** použije túto hodnotu pomocou príkazu **aclput**. **Poznámka:** Atribúty SUID, SGID a SVTX sa musia zhodovať s atribútmi, ktoré sú špecifikované pre režim, ak sú prítomné.
class	Názov skupiny súborov. Tento atribút povoľuje kontrolu niekoľkých súborov s rovnakým názvom triedy zadaním jedného argumentu pre príkaz **tcbck**. Je možné určiť viacero tried, pričom každá trieda musí byť oddelená čiarkou.
group	ID skupiny alebo názov skupiny súborov. Ak sa nezhoduje so skupinou súborov, príkaz **tcbck** nastaví ID skupiny súboru na túto hodnotu.
links	Zoznam čiarkou oddelených názvov ciest prepojených na tento súbor. Ak niektorý názov cesty v tomto zozname nie je prepojený na súbor, príkaz **tcbck** toto prepojenie vytvorí. Ak sa použije bez parametra *tree*, príkaz **tcbck**vytlačí správu o existencii nadbytočných pripojení, ale nekonkretizuje ich názvy. Ak sa použije s parametrom *tree*, príkaz **tcbck** vytlačí aj všetky dodatočné názvy ciest, ktoré sú k tomuto súboru pripojené.
mode	Zoznam čiarkou oddelených hodnôt. Povoliteľnými hodnotami sú SUID, SGID, SVTX a TCB. Prístupové práva pre súbor musia byť poslednou hodnotou a je ich možné zadať buď ako osmičkovú hodnotu, alebo ako reťazec pozostávajúci z 9 znakov. Napríklad hodnota 755 alebo rwxr-xr-x predstavuje platné prístupové práva pre súbor. Ak sa táto hodnota nezhoduje so skutočným režimom súboru, príkaz **tcbck** aplikuje správnu hodnotu.
owner	ID užívateľa alebo meno vlastníka súboru. Ak sa táto hodnota nezhoduje s vlastníkom súboru, príkaz **tcbck** nastaví ID vlastníka súboru na túto hodnotu.
program	Zoznam čiarkou oddelených hodnôt. Prvá hodnota je názov cesty programu vykonávajúceho kontrolu. Ďalšie hodnoty budú odovzdané programu ako argumenty, keď bude program spustený. **Poznámka:** Podľa toho, s akým príznakom bol príkaz **tcbck**použitý, prvý argument bude vždy niektorý z argumentov *-y*, *-n*, *-p* alebo *-t*.
source	Názov súboru, z ktorého sa má tento zdrojový súbor skopírovať pred vykonaním kontroly. Ak túto hodnotu nezadáte a ide o regulárny súbor, adresár alebo pomenovaný dátovod, vytvorí sa nová prázdna verzia tohto súboru (ak už neexistuje). Pri súboroch zariadenia platí, že sa pre zariadenie rovnakého typu vytvorí nový špeciálny súbor.
symlinks	Zoznam čiarkou oddelených názvov ciest symbolicky prepojených na tento súbor. Ak ľubovoľný názov cesty v zozname nie je symbolickým prepojením na súbor, príkaz **tcbck** symbolické prepojenie vytvorí. Ak bol príkaz **tcbck** použitý s argumentom *tree*, potom vytlačí aj všetky dodatočné názvy ciest, ktoré sú symbolickými odkazmi na tento súbor .

Ak v sekcii súboru /etc/security/sysck.cfg nie je uvedený atribút, príslušná kontrola sa nevykoná.

Používanie príkazu tcbck:

Príkaz **tcbck** sa používa na zabezpečenie správnej inštalácie súboru týkajúceho sa bezpečnosti, na zabezpečenie toho, aby strom súborového systému neobsahoval žiadne súbory, ktoré zjavne narúšajú systémovú bezpečnosť a na aktualizáciu, pridávanie alebo vymazávanie dôveryhodných súborov.

Príkaz **tcbck** sa bežne používa pre nasledujúce úlohy:

- Zabezpečenie správnej inštalácie súborov relevantných z pohľadu bezpečnosti.
- Kontrolu, že strom súborového systému neobsahuje žiadne súbory, ktoré evidentne narušujú bezpečnosť systému.
- Aktualizáciu, pridanie alebo odstránenie dôveryhodných súborov.

Príkaz **tcbck** možno používať nasledovnými spôsobmi:

- Normálne použitie
  - Neinteraktívna inicializácia systému
  - S príkazom **cron**.
- Interaktívne použitie
  - Preverte jednotlivé súbory a triedy súborov
- Paranoidné použitie
  - Súbor sysck.cfg uložíte offline a budete ho pravidelne obnovovať za účelom kontroly počítača.

Hoci z hľadiska šifrovanie nie je bezpečný, TCB používa príkaz **sum** pre kontrolné súčty. Databáza TCB sa dá nastaviť manuálne pomocou iného príkazu kontrolného súčtu, napríklad, príkaz **md5sum**, ktorý sa dodáva v balíku textutils RPM Package Manager na CD-disku *AIX Toolbox for Linux Applications CD*.

Kontrola dôveryhodných súborov:

Na kontrolu a opravu všetkých súborov v databáze tcbck a na opravu a vytvorenie protokolu všetkých chýb použite príkaz **tcbck**.

Ak chcete skontrolovať všetky súbory v databáze tcbck a opraviť a nahlásiť všetky chyby, napíšte:

```
tcbck -y ALL
```

To spôsobí, že príkaz **tcbck** skontroluje inštaláciu každého súboru v databáze tcbck, ktorú popisuje súbor /etc/security/sysck.cfg.

Ak chcete, aby sa toto vykonalo automaticky pri inicializácii systému a vytvoriť protokol o chybách, pridajte predchádzajúci reťazec príkazu do príkazu **/etc/rc**.

Kontrola stromu súborového systému:

Vždy keď máte podozrenie, že celistvosť systému mohla byť oslabená, spustite príkaz **tcbck**, aby skontroloval strom súborového systému.

Ak chcete skontrolovať strom súborového systému, napíšte:

```
tcbck -t tree
```

Keď bude príkaz **tcbck** vydaný s hodnotou *tree*, skontroluje sa správnosť inštalácie všetkých súborov v systéme (môže to trvať dlho). Ak príkaz **tcbck** zistí ľubovoľné súbory predstavujúce potenciálnu hrozbu z pohľadu bezpečnosti systému, môžete podozrivé súbory pozmeniť tak, že sa odstránia inkriminované atribúty. Okrem toho sa na všetkých ostatných súboroch v súborovom systéme vykonajú nasledujúce kontroly:

- Ak je vlastníkom súboru užívateľ s oprávneniami typu root a súbor má sadu bitov SetUID, bit SetUID sa vyčistí.
- Ak je skupina súborov administračnou skupinou, súbor je spustiteľný a súbor má sadu bitov SetGID, bit SetGID sa vyčistí.

- Ak má súbor nastavený atribút **tcb**, tento atribút sa vymaže.
- Ak je súbor zariadením (znakový alebo blokový špeciálny súbor), bude odstránený.
- Ak je súbor dodatočným prepojením na názov cesty popísaný v súbore /etc/security/sysck.cfg, prepojenie sa odstráni.
- Ak je súbor dodatočným symbolickým prepojením na názov cesty popísaný v súbore /etc/security/sysck.cfg, symbolické prepojenie sa odstráni.

  Poznámka: Všetky položky zariadení museli byť do súboru /etc/security/sysck.cfg pridané ešte pred spustením príkazu **tcbck** inak bude systém vyhlásený za nepoužiteľný. Ak chcete dôveryhodné zariadenia pridať do súboru /etc/security/sysck.cfg, použite príznak **-l**.

Upozornenie: *Ne*spúšťajte voľbu príkazu **tcbck -y tree**. Táto voľba odstráni a zakáže všetky zariadenia, ktoré nie sú správne uvedené v TCB a môže znefunkčniť systém.

Pridanie dôveryhodného programu:

Použite príkaz **tcbck** na pridávanie špecifického programu do súboru /etc/security/sysck.cfg.

Ak chcete pridať konkrétny program do súboru /etc/security/sysck.cfg, zadajte:

```
tcbck -a PathName [Attribute=Value]
```

V príkazovom riadku je nutné zadať len atribúty, ktorých hodnoty nie sú odvodené z aktuálneho stavu súboru. Názvy všetkých atribútov sú obsiahnuté v súbore /etc/security/sysck.cfg.

Napríklad nasledovný príkaz registruje nový program SetUID root pomenovaný /usr/bin/setgroups, ktorý má prepojenie nazvané /usr/bin/getgroups:

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

Ak chcete jfh a jsl pridať ako administračných užívateľov a developers pridať ako administračnú skupinu, ktorá sa má overiť počas auditu bezpečnosti súboru /usr/bin/abc, napíšte:

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

Po nainštalovaní programu nemusíte vedieť, ktoré nové súbory boli registrované v súbore /etc/security/sysck.cfg. Tieto súbory možno nájsť a pridať nasledovným príkazom:

```
tcbck -t tree
```

Tento príkaz zobrazí názov ľubovoľného súboru, ktorý sa má registrovať v súbore /etc/security/sysck.cfg.

Vymazanie dôveryhodného programu:

Ak zo systému odstránite súbor, ktorý je popísaný v súbore /etc/security/sysck.cfg, musíte odstrániť aj popis tohto súboru zo súboru /etc/security/sysck.cfg.

Napríklad, ak ste vymazali program /etc/cvid, nasledujúci reťazec príkazu zapríčiní chybovú správu:

```
tcbck -t ALL
```

Výsledná chybová správa bude:

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

Popis pre tento program zostáva v súbore /etc/security/sysck.cfg. Ak chcete odstrániť popis tohto programu, zadajte nasledovný príkaz:

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

Konfigurácia ďalších dôveryhodných volieb:

Môžete nakonfigurovať ďalšie voľby pre Trusted Computing Base (TCB).

Obmedzenie prístupu na terminál:

Môžete nakonfigurovať operačný systém na obmedzenie prístupu na terminál.

Príkazy **getty** a **shell** slúžia na zmenu vlastníka a režimu terminálu, čím bránia nedôveryhodným programom pristupovať k terminálu. Operačný systém umožňuje konfigurovať exkluzívny prístup k terminálu.

Používanie Secure Attention Key:

Dôveryhodná komunikačná cesta sa vytvorí po stlačení vyhradenej kombinácie klávesov (Ctrl-X, a potom Ctrl-R) s názvom SAK (Secure Attention Key) .

Poznámka: SAK používajte opatrne, pretože zastavuje všetky procesy, ktoré sa snažia o prístup na terminál a všetky odkazy naň (napríklad /dev/console možno pripojiť k /dev/tty0).

Dôveryhodná komunikačná cesta sa vytvorí za nasledovných podmienok:

- Pri prihlasovaní do systému

  Po stlačení klávesov funkcie SAK:
  - Ak sa zobrazí nová prihlasovacia obrazovka, máte zabezpečenú cestu.
  - Ak sa zobrazí výzva dôveryhodného prostredia, úvodná prihlasovacia obrazovka bola neautorizovaným programom, ktorý sa mohol pokúšať o získanie vášho hesla. Zistite, kto aktuálne používa terminál použitím príkazu **who** a potom sa odhláste.
- Keď chcete, aby zadaný príkaz spustil dôveryhodný program. Niekoľko príkladov:
  - Spustenie, keď ste užívateľ s oprávneniami typu root. Ako užívateľ s oprávneniami typu root spúšťajte programy len po vytvorení dôveryhodnej komunikačnej cesty. Zabezpečíte tak, že nedôveryhodné programy sa nebudú spúšťať s oprávnením užívateľa typu root.
  - Spustenie príkazov **su**, **passwd** a **newgrp**. Tieto príkazy spúšťajte len po vytvorení dôveryhodnej komunikačnej cesty.

Konfigurácia Secure Attention Key:

Secure Attention Key sa konfiguruje, aby sa vytvorila dôveryhodná komunikačná cesta.

Každý terminál možno nakonfigurovať nezávisle, takže po stlačení klávesy funkcie Secure Attention Key (SAK) na tomto termináli sa vytvorí dôveryhodná komunikačná cesta. Toto určuje atribút **sak_enabled** v súbore /etc/security/login.cfg. Ak je hodnota tohto atribútu True, funkcia SAK je povolená.

Ak sa má na komunikáciu použiť port, (napríklad, pomocou príkazu **uucp**), konkrétne použitý port bude mať v stati súboru /etc/security/login.cfg nasledujúci riadok:

```
sak_enabled = false
```

Tento riadok (alebo žiadna zadaná hodnota v tejto stati) zakáže funkciu SAK pre takýto terminál.

Ak chcete povoliť SAK na termináli, do časti pre tento terminál pridajte nasledovný riadok:

```
sak_enabled = true
```

Trusted Execution

Trusted Execution (TE) je sada funkcií, ktorá slúži na overenie integrity systému a implementáciu rozšírených bezpečnostných politík, ktoré zvyšujú úroveň dôveryhodnosti celého systému.

Zlomyseľný užívateľ, ktorý chce poškodiť systém, sa obvykle snaží získať prístup do systému a nainštalovať tam trójskeho koňa alebo rootkit, prípadne odkryť kľúčové súbory z hľadiska bezpečnosti, čo privodí zraniteľnosť alebo zneužiteľnosť systému. Ústredný koncept v pozadí sady nástrojov Trusted Execution je prevencia takýchto úkonov, alebo, v horšom prípade, aspoň ich odhalenie, keď sa vyskytnú v systéme. Vďaka funkciám Trusted Execution môže administrátor systému určiť presnú sadu spustiteľných programov, ktoré sa môžu spustiť, alebo sadu rozšírení kernelu, ktoré sa môžu zaviesť do pamäte. Môžete ich použiť aj na audit bezpečnosti systému a identifikáciu súborov, ktoré boli zmenené, a zvýšiť tak úroveň dôveryhodnosti systému, takže bude ťažšie ho ohroziť. Sada vlastností v rámci TE sa dá rozdeliť do týchto skupín:

- Správa databázy dôveryhodných podpisov (TSD)
- Audit integrity databázy dôveryhodných podpisov (TSD)
- Konfigurácia bezpečnostných politík
- Dôveryhodná cesta spustenia (TEP) a dôveryhodná cesta knižnice (TLP)

Poznámka: V operačnom systéme AIX už existuje mechanizmus TCB. TE je silnejší a komplexnejší mechanizmus, ktorý presahuje možnosti TCB a poskytuje rozšírené bezpečnostné politiky na lepšie riadenie integrity systému. Kým je Trusted Computing Base stále dostupný, Trusted Execution predstavuje nový a pokročilejší koncept na overovanie a zabezpečenie integrity systému.

Riadenie databázy dôveryhodných podpisov:

Podobne ako Trusted Computing Base (dôveryhodná výpočtová základňa) existuje databáza, v ktorej sú uložené kľúčové bezpečnostné parametre dôveryhodných súborov, nachádzajúcich sa v systéme. Táto databáza s názvom Trusted Signature Database (TSD), sa nachádza v súbore /etc/security/tsd/tsd.dat.

Dôveryhodný súbor je súbor, ktorý je kritický s hľadiska bezpečnosti systému, a jeho odhalenie môže ohroziť bezpečnosť celého systému. Tomuto opisu typicky zodpovedajú nasledujúce súbory:

- Kernel (operačný systém)
- Všetky programy koreňového setuid
- Všetky programy koreňového setgid
- Akýkoľvek program, ktorý spúšťa výlučne koreňový užívateľ alebo člen systémovej skupiny
- Akýkoľvek program, ktorý musí spustiť administrátor, kým je na dôveryhodnej ceste spustenia (napríklad príkaz **ls**)
- Konfiguračné súbory, ktoré riadia systémové operácie
- Akýkoľvek program spustený s privilégiom alebo prístupovými právami na zmenu kernelu alebo konfiguračných súborov systému

Každý dôveryhodný súbor by mal ideálne mať priradený odsek alebo definíciu súboru, uloženú v databáze dôveryhodných podpisov (TSD). Súbor môže byť označený ako dôveryhodný tým, že jeho definíciu pridáte do databázy TSD pomocou príkazu **trustchk**. Príkaz **trustchk** môžete použiť na pridanie, vymazanie alebo vypísanie položiek z databázy TSD.

Databáza dôveryhodných podpisov:

Databáza dôveryhodných podpisov (TSD) je databáza, v ktorej sú uložené kľúčové bezpečnostné parametre dôveryhodných súborov, nachádzajúcich sa v systéme. Táto databáza sa nachádza v adresári /etc/security/tsd/tsd.dat.

K všetkým dôveryhodným súborom musí byť v ideálnom prípade priradený odsek alebo definícia súboru v databáze Trusted Signature Database (TSD). Každý dôveryhodný súbor má priradenú jedinečnú hašovaciu hodnotu a digitálny podpis. Šifrovacia hašovacia hodnota predvolenej množiny dôveryhodných súborov sa generuje s použitím algoritmu SHA-256 a digitálneho podpisu, ktorý generuje zostavovacie prostredie AIX s použitím algoritmu RSA a je súčasťou inštalačných sád súborov operačného systému AIX. Tieto hašovacie hodnoty a podpisy sú doručené ako súčasť príslušných inštalačných obrazov AIX a sú uložené v databáze dôveryhodného softvéru (/etc/security/tsd/tsd.dat) na cieľovom počítači, vo formáte odsekov ako v nasledujúcej ukážke:

```
/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 Vlastník súboru. Túto hodnotu vypočíta príkaz **trustchk**, keď sa súbor pridáva do TSD.

group Skupina súborov. Túto hodnotu vypočíta príkaz **trustchk**.

mode Zoznam čiarkou oddelených hodnôt. Prípustné hodnoty sú **SUID** (bit nastavenia SUID), **SGID** (bit nastavenia SGID), **SVTX** (bit nastavenia SVTX) a **TCB** (dôveryhodná výpočtová základňa). Prístupové práva pre súbor musia byť poslednou hodnotou a je ich možné zadať ako osmičkovú hodnotu. Napríklad, v prípade súboru, ktorý je určený identifikátorom **uid** a má bity oprávnení **rwxr-xr-x**, je hodnota režimu **SUID, 755**. Túto hodnotu vypočíta príkaz **trustchk**.

type Typ súboru. Túto hodnotu vypočíta príkaz **trustchk**. Možné hodnoty sú **FILE**, **DIRECTORY**, **MPX_DEV**, **CHAR_DEV**, **BLK_DEV** a **FIFO**.

hardlinks
Zoznam pevných odkazov na súbor. Táto hodnota sa nedá vypočítať príkazom **trustchk**. Musí ju dodať užívateľ, keď súbor pridáva do databázy.

symlinks
Zoznam symbolických odkazov na súbor. Táto hodnota sa nedá vypočítať príkazom **trustchk**. Musí ju dodať užívateľ, keď súbor pridáva do databázy.

size Definuje veľkosť súboru. Hodnota **VOLATILE** znamená, že súbor sa často mení.

cert_tag
Toto pole mapuje digitálny podpis súboru na súvisiaci certifikát, ktorý sa môže použiť na overenie podpisu súboru. Do tohto poľa sa ukladá ID certifikátu, ktoré vypočítava príkaz **trustchk** pri pridaní súboru do databázy TSD. Tieto certifikáty sú uložené v adresári /etc/security/certificates.

signature
Digitálny podpis súboru. Hodnota **VOLATILE** znamená, že súbor sa často mení. Toto pole vypočíta príkaz **trustchk**.

hash_value
Šifrovacia hašovacia hodnota súboru. Hodnota **VOLATILE** znamená, že súbor sa často mení. Toto pole vypočíta príkaz **trustchk**.

minslabel
Definuje minimálne označenie citlivosti pre objekt.

maxslabel
Definuje návestie maximálnej citlivosti pre objekt (platí v dôveryhodnom systéme AIX). Tento atribút sa nedá použiť na normálne súbory a fifo.

intlabel
Definuje návestie integrity pre objekt (platí v dôveryhodnom systéme AIX).

accessauths
Definuje autorizáciu prístupu na objekte (platí v dôveryhodnom systéme AIX).

innateprivs
Definuje vlastné privilégiá pre súbor.

proxyprivs
Definuje proxy privilégiá pre súbor.

authprivs
Definuje privilégiá, ktoré sú priradené užívateľovi po danej autorizácii.

secflags
Definuje príznaky zabezpečenia súboru, priradené objektu.

t_accessauth
Definuje ďalší dôveryhodný systém AIX s autorizáciami prístupu špecifickými pre viacúrovňovú bezpečnosť (MLS) (platí v dôveryhodnom systéme AIX).

t_innateprivs
Definuje ďalšie prostredie Trusted AIX so zdedenými privilégiami MLS pre súbor (platné pre systém Trusted AIX).

t_proxyprivs
Definuje ďalšie prostredie Trusted AIX s privilégiami proxy MLS pre súbor (platné pre systém Trusted AIX).

t_authprivs
Definuje ďalšie prostredie Trusted AIX s privilégiami MLS, ktoré sú priradené k užívateľovi po daných oprávneniach (platné pre systém Trusted AIX).

t_secflags
Definuje ďalšie prostredie Trusted AIX s príznakmi bezpečnosti súborov MLS priradenými k objektu (platné pre systém Trusted AIX).

Keď pridáte novú položku do databázy TSD a na dôveryhodný súbor sa odkazujú symbolické alebo pevné prepojenia, tieto prepojenia môžete pridať do databázy TSD prostredníctvom atribútov **symlinks** a **hardlinks** príkazu **trustchk** z príkazového riadka. Ak očakávate, že pridávaný súbor sa bude často meniť, potom z príkazového riadka spustite príkaz VOLATILE. Príkaz **trustchk** tak nebude počítať polia **hash_value** a **signature** pri generovaní definície súboru, ktorá sa pridá do TSD. Počas overovania integrity tohto súboru budú polia **hash_value** a **signature** ignorované.

Počas pridávania definícií regulárnych súborov do TSD je potrebné poskytnúť súkromný kľúč (vo formáte ASN.1/DER). Použite príznak **-s** a digitálny certifikát s príslušným verejným kľúčom použitím príznaku **-v**. Súkromný kľúč sa použije na vygenerovanie podpisu súboru a následne sa odstráni. Je úlohou užívateľa bezpečne kľúč uložiť. Certifikát sa uloží v sklade certifikátov do súboru /etc/security/certificates, aby sa podpisy dali overiť vždy, keď si vyžiadate overenie integrity. Keďže kalkulácia podpisu nie je možná pre neregulárne súbory, ako sú súbory zariadení a adresáre, nie je povinné poskytnutie súkromného kľúča a certifikátu počas pridávania takýchto súborov do TSD.

Taktiež môžete poskytnúť vopred vypočítanú definíciu súboru prostredníctvom súboru, ktorý sa má pridať do databázy TSD, s použitím voľby **-f**. V tomto prípade príkaz **trustchk** nevypočíta žiadne hodnoty a definície uloží do databázy TSD bez overovania. V tomto prípade za zdravie definícií súborov zodpovedá užívateľ.

Podpora overovania knižníc

Pre podporu overovania knižníc sa súbor tsd.dat pridá do adresára /etc/security/tsd/lib/. Názov databázy je /etc/security/tsd/lib/lib.tsd.dat. Táto databáza je určená špeciálne pre knižnice zahŕňajúce odseky pre súbory .o príslušnej dôveryhodnej knižnice. Odseky pre jednotlivé súbory .o knižnice majú formát uvedený v nasledujúcom príklade.

V prípade knižnice libc.a, ak má súbor strcmp.o typ súboru .o, odsek pre súbor strcmp.o v adresári /etc/security/tsd/lib/lib.tsd.dat bude podobný odseku v nasledujúcom príklade:

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

Táto databáza obsahuje položky prislúchajúce k atribútom **type**, **size hash**, **cert tag** a **signature** súboru .o. Hašovacia hodnota knižnice sa aktualizuje v súbore /etc/security/tsd/tsd.dat pre príslušný odsek. Tieto hodnoty atribútov sa dynamicky generujú počas zostavovania a hodnoty sa presúvajú do databázy /etc/security/tsd/lib/lib.tsd.dat počas inštalácie.

V súbore /etc/security/tsd/tsd.dat sa odseky knižníc upravia tak, aby uvádzali atribút **type** ako LIB a atribúty **size** a **signature** ako prázdne hodnoty. V súčasnosti sa hodnoty **dynamických** atribútov **size**, **hash**, **signature** uchovávajú ako hodnota **VOLATILE**. Preto sa overovanie knižníc preskočí počas zavádzania systému. Počnúc vydaním AIX 6.1.0, sa hodnoty **size**, **hash** a **signature** odsekov dôveryhodných knižníc vypočítajú pomocou súborov .o knižnice. Počas inštalácie sa databáza tsd.dat naplní hodnotami uvádzajúcimi vypočítané hodnoty a príslušné odseky súborov .o dôveryhodných knižníc sa uložia do databázy /etc/security/tsd/lib/lib.tsd.dat.

Vzdialený prístup k databáze TE:

Centralizované politiky TSD (Trusted Signature Database) a TE (Trusted Execution) možno do systémového prostredia implementovať ich uložením v LDAP.

Databázy, ktoré riadia politiky TSD a politiky TE, sú uložené samostatne pre jednotlivé systémy. AIX Centralizované politiky TSD a TE sú uložené v adresári LDAP, aby ich bolo možné spravovať centrálne. Používanie centralizovaných politík TSD a TE umožňuje kontrolovať, či sú politiky v LDAP hlavnou kópiou a či môžu politiky aktualizovať klientov vždy, keď je klient znova nainštalovaný, aktualizovaný alebo keď sa poruší bezpečnosť. Centralizované politiky TE umožňujú jedno umiestnenie na vynútenie politík TE bez potreby aktualizovať každého klienta osobitne. Centralizované politiky TSD sa oveľa ľahšie riadia ako politiky TSD, ktoré nie sú centralizované.

AIX Na exportovanie údajov lokálnych politík TSD a TE na servery LDAP, konfigurovanie klientov na používanie údajov politík TSD a TE na serveroch LDAP, riadenie vyhľadávania údajov politík TSD a TE a správu údajov LDAP z klientskeho systému môžete použiť pomocné programy. Nasledujúce časti poskytujú bližšie informácie o týchto funkciách.

Export údajov politík TSD a TE do LDAP:

Ak chcete používať LDAP ako centralizovaný archív pre politiky TSD a TE, LDAP server musí byť zaplnený údajmi politiky.

Skôr ako klienti LDAP budú môcť použiť server na údaje politiky, LDAP server musí mať nainštalovanú schému politík TSD a TE. Schéma politík TSD a TE pre LDAP je k dispozícii v systéme AIX v súbore **/etc/security/ldap/sec.ldif** . Schéma pre LDAP server musí byť aktualizovaná týmto súborom pomocou príkazu **ldapmodify**.

Ak chcete identifikovať verziu databáz TE na LDAP serveri a informovať o nej klientov LDAP, musíte nastaviť atribút **databasename** v súbore **/etc/nscontrol.conf**. Atribút **databasename** berie ľubovoľný názov ako hodnotu a používa ho príkaz **tetoldif** počas generovania formátu ldif.

Pomocou príkazu **tetoldif** prečítajte údaje v lokálnych súboroch politík TSD a TE a pripravte výstup politík vo formáte, ktorý možno použiť pre LDAP. Výstup vygenerovaný príkazom **tetoldif** možno uložiť do súboru vo formáte ldif a následne použiť na zaplnenie LDAP servera údajmi pomocou príkazu **ldapadd**. Príkaz **tetoldif** používa nasledujúce databázy v lokálnom systéme na vygenerovanie údajov politík TSD a TE pre LDAP:

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

Konfigurácia klienta LDAP pre politiky TSD a TE:

Systém musí byť nakonfigurovaný ako klient LDAP, aby mohol používať údaje politík TSD a TE uložené v LDAP.

Pomocou príkazu AIX **/usr/sbin/mksecldap** nakonfigurujte systém ako klienta LDAP. Príkaz **mksecldap** dynamicky vyhľadá označený LDAP server, aby zistil umiestnenie údajov politík TSD a TE, a výsledky uloží do súboru **/etc/security/ldap/ldap.cfg**.

Po úspešnej konfigurácii systému ako klienta LDAP pomocou príkazu **mksecldap** musí byť systém ďalej konfigurovaný, aby povolil LDAP ako doménu vyhľadávania pre údaje politík TSD a TE konfiguráciou atribútu secorder súboru **/etc/nscontrol.conf**.

Démon klienta **/usr/sbin/secldapclntd** po nakonfigurovaní systému ako klienta LDAP a ako domény vyhľadávania pre údaje politík TSD a TE získa údaje politík TSD a TE z LDAP servera vždy, keď sa v klientovi LDAP vykoná ľubovoľný príkaz **trustchk**.

Aktivácia LDAP pomocou príkazu trustchk:

Všetky príkazy riadenia databázy politík TSD a TE sú povolené na používanie databázy politík TSD a TE LDAP.

Pomocou príkazu **trustchk** s návesťou **–R** vykonajte úvodné nastavenie databázy LDAP. Úvodné nastavenie zahŕňa pridanie politík TSD, politík TE, základných charakteristických názvov a vytvorenie súboru **/etc/security/tsd/ldap/tsd.dat** a súboru **/etc/security/tsd/ldap/tepolicies.dat** lokálnej databázy.

Ak je príkaz **trustchk** spustený s návesťou **–R** pomocou voľby LDAP, operácie budú založené na údajoch LDAP servera. Ak je príkaz **trustchk** spustený s návesťou **–R** pomocou voľby files, operácie budú založené na údajoch lokálnej databázy. Predvolená hodnota pre návesť **–R** je použiť voľbu files.

Súvisiace informácie:

príkaz mksecldap

príkaz trustchk

Audit integrity databázy dôveryhodných podpisov:

Pomocou príkazu **trustchk** môžete vykonať audit stavu integrity definícií súborov v databáze dôveryhodných podpisov (TSD) a porovnať ich so skutočnými súbormi.

Ak príkaz **trustchk** zistí nejakú anomáliu, môže ju automaticky opraviť, alebo sa pred pokusom vykonať opravu opýta užívateľa. V prípade anomálií typu nezhoda veľkosti, podpisu, cert_tag alebo hash_value oprava nie je možná. V takom prípade príkaz **trustchk** daný súbor zneprístupní, čím sa stane nepoužiteľným.

Pre rôzne nezhodujúce sa atribúty je potrebné vykonať nasledujúce nápravné kroky:

owner Vlastník súboru by mal byť zmenený na hodnotu v TSD.

group Skupina alebo súbor by mal byť zmenený na hodnotu v TSD.

mode Bity režimu súboru by mali byť zmenené na hodnotu v TSD.

hardlinks
Ak odkaz ukazuje na nejaký iný súbor, upraví sa tak, aby ukazoval na tento súbor. Ak odkaz neexistuje, vytvorí sa nový odkaz ukazujúci na súbor.

symlinks
Rovnako ako pri hardlinks.

type Súbor sa zneprístupní.

size Súbor sa zneprístupní, s výnimkou súboru **VOLATILE**.

cert_tag
Súbor sa zneprístupní.

signature
Súbor sa zneprístupní, s výnimkou súboru **VOLATILE**.

hash_value
Súbor sa zneprístupní, s výnimkou súboru **VOLATILE**.

minslabel
V systéme Trusted AIX sa značka minimálnej citlivosti zmení na hodnotu v TSD.

maxslabel
V systéme Trusted AIX sa značka maximálnej citlivosti zmení na hodnotu v TSD.

intlabel
V systéme Trusted AIX sa značka integrity zmení na hodnotu v TSD.

accessauths
Prístupové oprávnenia sa zmenia na hodnotu v TSD. V systéme Trusted AIX sa hodnoty **t_accessauths** považujú za súčasť atribútu **accessauths**.

innateprivs
Vlastné privilégiá sa zmenia na hodnotu v TSD. V systéme Trusted AIX sa hodnoty **t_innateprivs** považujú za súčasť atribútu **innateprivs**.

inheritprivs
Dediteľné privilégiá sa zmenia na hodnotu v TSD. V systéme Trusted AIX sa hodnoty **t_inheritprivs** považujú za súčasť atribútu inherit.

authprivs
Autorizované privilégiá sa zmenia na hodnotu v TSD. V systéme Trusted AIX sa hodnoty **t_authprivs** považujú za súčasť atribútu **authprivs**.

aecflags
Príznaky zabezpečenia sa zmenia na hodnotu v TSD. V systéme Trusted AIX sa hodnoty **t_secglags** považujú za súčasť atribútu **secflags**.

Platnosť definícií súborov môžete tiež overiť porovnaním s inou databázou pomocou voľby **-F**. Administrátor systému by nemal databázu TSD uložiť na ten istý systém a zálohovať ju na inom umiestnení. Integrita súborov sa dá porovnať so zálohovanou verziu databázy TSD s použitím voľby **-F**.

Konfigurácia bezpečnostných politík:

Funkcia Trusted Execution (TE) vám ponúka mechanizmus runtime overovania integrity súborov. Pomocou tohto mechanizmu môžete nakonfigurovať systém tak, aby kontroloval integritu dôveryhodných súborov pred každou požiadavkou na prístup k týmto súborom, takže na systéme bude možné pristúpiť jedine k dôveryhodným súborom, ktoré prešli kontrolou integrity.

Keď je súbor označený ako dôveryhodný (pridaním jeho definície do databázy dôveryhodných podpisov - TSD), funkcia TE môže monitorovať jeho integritu pri každom prístupe. TE môže nepretržite monitorovať systém a dokáže odhaliť nedovolenú manipuláciu s ľubovoľným dôveryhodným súborom (zo strany škodlivej aplikácie alebo

užívateľa), ktorý sa nachádza v systéme počas runtime (napríklad počas zavádzania). Ak sa zistí, že sa s nejakým súborom manipulovalo, TE môže vykonať nápravné kroky na základe predkonfigurovaných politík, napríklad nepovoliť spustenie, prístup k súboru, alebo zaprotokolovanie chyby. Ak má otváraný alebo spúšťaný súbor položku v databáze dôveryhodných podpisov (TSD), TE postupuje nasledovne:

- Pred načítaním binárneho súboru komponent zodpovedný za načítanie súboru (systémový zavádzač) vyvolá podsystém Trusted Execution a vypočíta hašovaciu hodnotu pomocou algoritmu SHA-256 (konfigurovateľný).
- Táto vypočítaná runtime hodnota sa porovná s hodnotou uloženou v TSD.
- Ak sa hodnoty zhodujú, povolí sa otvorenie alebo spustenie súboru.
- Ak sa nezhodujú, znamená to, že binárny súbor bol zmanipulovaný, alebo inak ohrozený. Je na užívateľovi rozhodnúť, aký postup sa má zvoliť. Mechanizmus TE poskytuje voľby, ktoré užívateľom dovoľujú definovať si svoje vlastné politiky pre postupy v prípade, keď sa hašovacie hodnoty nezhodujú.
- Na základe konfigurovaných politík sa vykoná príslušná akcia.

Nakonfigurovať sa dajú nasledujúce politiky:

CHKEXEC
: Skontrolovať hašovaciu hodnotu iba dôveryhodných spustiteľných súborov pred ich zavedením do pamäte pri spustení.

CHKSHLIBS
: Skontrolovať hašovaciu hodnotu iba dôveryhodných zdieľaných knižníc pred ich zavedením do pamäte pri spustení.

CHKSCRIPTS
: Skontrolovať hašovaciu hodnotu iba dôveryhodných skriptov shell pred ich zavedením do pamäte.

CHKKERNEXT
: Skontrolovať hašovaciu hodnotu iba rozšírenia kernelu pred jeho zavedením do pamäte.

STOP_UNTRUSTD
: Zastaviť načítavanie súborov, ktoré nie sú dôveryhodné. Načítavať sa budú iba súbory patriace do databázy TSD. Táto politika funguje iba v kombinácii s niektorou politikou CHK* uvedenou vyššie. Napríklad ak platí **CHKEXEC=ON** a **STOP_UNTRUSTD=ON**, potom je zablokované iba spustenie spustiteľných binárnych súborov, ktoré nepatria do databázy TSD.

STOP_ON_CHKFAIL
: Zastaviť načítavanie dôveryhodných súborov, ktoré neprešli kontrolou hašovacej hodnoty. Táto politika funguje aj v kombinácii s politikami CHK*. Napríklad ak platí **CHKSHLIBS=ON** a **STOP_ON_CHKFAIL=ON**, potom bude zablokované načítanie do pamäte každej zdieľanej knižnice, ktorá nepatrí do TSD.

TSD_LOCK
: Zamknúť databázu TSD, takže nebude dostupné pre úpravy.

TSD_FILES_LOCK
: Zamknúť dôveryhodné súbory. Nedovolí otvoriť dôveryhodné súbory v režime zápisu.

TE Povoliť/zakázať funkcie Trusted Execution. Iba keď je táto možnosť povolená, sú vyššie uvedené politiky účinné.

Nasledujúca tabuľka uvádza interakcie medzi rôznymi politikami CHK* a STOP*, keď sú povolené:

Politika	STOP_UNTRUSTD	STOP_ON_CHKFAIL
CHKEXEC	Zastaviť načítavanie spustiteľných súborov, ktoré nepatria do TSD.	Zastaviť načítavanie spustiteľných súborov, ktorých hašovacie hodnoty sa nezhodujú s hodnotami v TSD.
CHKSHLIBS	Zastaviť načítavanie zdieľaných knižníc, ktoré nepatria do TSD.	Zastaviť načítavanie zdieľaných knižníc, ktorých hašovacie hodnoty sa nezhodujú s hodnotami v TSD.
CHKSCRIPTS	Zastaviť načítavanie skriptov shell, ktoré nepatria do TSD.	Zastaviť načítavanie skriptov shell, ktorých hašovacie hodnoty sa nezhodujú s hodnotami v TSD.
CHKKERNEXT	Zastaviť načítavanie rozšírení kernelu, ktoré nepatria do TSD.	Zastaviť načítavanie rozšírení kernelu, ktorých hašovacie hodnoty sa nezhodujú s hodnotami v TSD.

Poznámka: Politiku môžete kedykoľvek zapnúť alebo vypnúť, pokiaľ je TE zapnuté. Keď je politika aktívna, jej zakázanie bude účinné až pri nasledujúcom zavedení. Všetky informačné správy sa protokolujú do **syslog**.

Súvisiace informácie:

Služba jadra TE_verify_reg

Služba jadra TE_verify_unreg

Trusted Execution Path a Trusted Library Path:

Trusted Execution Path (TEP - dôveryhodná cesta spustenia) definuje zoznam adresárov, ktoré obsahujú dôveryhodné spustiteľné súbory. Keď je povolené overovanie TEP, systémový zavádzač umožňuje spustenie iba binárnych súborov na zadaných cestách. Trusted Library Path (TLP - dôveryhodná cesta knižnice) funguje rovnako, len definuje adresáre obsahujúce dôveryhodné knižnice v systéme.

Keď je povolené TLP, systémový zavádzač umožňuje vytvorenie pripojenia binárnych súborov iba ku knižniciam z tejto cesty. Príkaz **trustchk** sa dá použiť na povolenie aj zakázanie TEP alebo TLP, a tiež na nastavenie ciest, oddelených dvojbodkou, s použitím atribútov TEP alebo TLP v riadku príkazu **trustchk**.

Trusted Shell a Secure Attention Key:

Trusted Shell a Secure Attention Key (SAK) fungujú podobne ako dôveryhodná výpočtová základňa (TCB), až na to, že ak je v systéme namiesto TCB povolené Trusted Execution, Trusted Shell spúšťa len súbory, ktoré patria do databázy dôveryhodných podpisov.

Bližšie informácie o TCB a SAK nájdete v témach Dôveryhodná výpočtová základňa, Používanie Secure Attention Key a Konfigurácia Secure Attention Key.

Databáza politík TE (Trusted Execution):

Politiky TE (Trusted Execution) sú uložené v súbore **/etc/security/tsd/tepolicies.dat**. Cesta pre politiky TE je uvedená s adresármi TLP a TEP.

Security Profile Evaluation Assurance Level 4+ a Labeled AIX Security and Evaluation Assurance Level 4+

Administrátori systému môžu počas inštalácie základného operačného systému (BOS) nainštalovať systém s voľbou Base AIX Security (BAS) a Evaluation Assurance Level 4+ (EAL4+) alebo Labeled AIX Security (LAS) a Evaluation Assurance Level 4+ (EAL4+). Na systém, na ktorom boli vybraté tieto voľby, sa vzťahujú obmedzenia na inštaláciu softvéru počas inštalácie základného operačného systému, ako aj obmedzenia sieťového prístupu.

Poznámka: Pre AIX, verzia 7.1 evaluácie prebiehajú v súčasnosti. Najnovšie informácie nájdete v poznámkach k vydaniu pre AIX, verzia 7.1.

Prehľad produktu Security profile:

Security profile je produkt určujúci bezpečnostné požiadavky pre všeobecne použiteľné operačné systémy v sieťových prostrediach. Tento profil vytvorí požiadavky, potrebné na dosiahnutie bezpečnostných cieľov bezpečnostnej funkcie TOE (Target of evaluation) a jej prostredia.

Produkt Security obsahuje základný balík a niekoľko rozšírených balíkov. Produkty súvisiace s podporou základného balíka produktu Security profile sú Identification and Authentication, Discretionary Access Control (DAC), Auditing, Cryptographic Services, Management of Security Mechanisms, a Trusted Channel communications. Produkt Security profile zahŕňa ďalšie, voliteľné balíky pre bezpečnosť založenú na návestiach, overovanie integrity, rozšírené auditovanie, všeobecné šifrovanie, rozšírenú správu, rozšírenú identifikáciu a autentifikáciu, dôveryhodné zavedenie a virtualizáciu.

Predpoklady

- Prostredie, ktoré sa má použiť pre funkciu TOE:

  Všetky predpoklady v tejto sekcii odkazujú na Base AIX Security (režim BAS) a Labeled AIX Security (režim LAS), pokiaľ nie je stanovené inak. Všetky predpoklady, súvisiace so serverom VIOS (Virtual input output server), sú explicitne označené len ako VIOS. VIOS nezdieľa predpoklady ani s operačným systémom AIX, ani s prostredím Trusted AIX.
- Fyzická:

  Prostredie IT poskytuje funkciu TOE s príslušnou fyzickou bezpečnosťou, ktorá zodpovedá hodnote aktív IT, chránených funkciou TOE.

  Poznámka: Len VIOS: Operačné prostredie poskytuje funkciu TOE s príslušnou fyzickou bezpečnosťou, ktorá zodpovedá hodnote aktív IT, chránených funkciou TOE.
- Administrácia:
  - Bezpečnostná funkcia TOE je riadená jedným alebo viacerými kompetentnými jednotlivcami. Personál administrácie systému nie je neopatrný, úmyselne nedbanlivý alebo nepriateľsky zaujatý a dodržiava pokyny, uvedené v riadiacej dokumentácii.
  - Autorizovaní užívatelia môžu pristupovať k niektorým informáciám, riadeným funkciou TOE, a očakáva sa od nich, že budú spolupracovať.
  - Užívatelia sú dostatočne vyškolení a dôveryhodní, aby mohli vykonávať niektorú úlohu alebo skupinu úloh v bezpečnom prostredí IT. Musia vykonávať kompletnú kontrolu nad svojimi užívateľskými údajmi.
  - Len VIOS: Bezpečnostná funkcia TOE je riadená jedným alebo viacerými kompetentnými jednotlivcami. Personál administrácie systému nie je neopatrný, úmyselne nedbanlivý alebo nepriateľsky zaujatý a dodržiava pokyny, uvedené v riadiacej dokumentácii.
  - Len VIOS: Autorizovaní užívatelia vlastnia potrebnú autorizáciu na prístup aspoň k niektorým informáciám, riadeným funkciou TOE, a očakáva sa od nich, že budú spolupracovať.
  - Len VIOS: Užívatelia sú dostatočne vyškolení a dôveryhodní, aby mohli vykonávať niektorú úlohu alebo skupinu úloh v bezpečnom operačnom prostredí. Musia vykonávať kompletnú kontrolu nad svojimi užívateľskými údajmi.
- Procedurálne:
  - Každá úprava alebo poškodenie súborov funkcie TOE, uplatňujúcich bezpečnosť alebo podstatných pre bezpečnosť, ktorú spôsobil užívateľ alebo základná platforma úmyselne alebo náhodne, musí byť zistená administrátorom.
  - Predpokladá sa, že všetky vzdialené dôveryhodné systémy IT, ktorým dôveruje funkcia TSF (Target Security Function) a ktoré majú poskytovať údaje alebo služby TSF funkcii TOE alebo majú podporovať TSF v uplatňovaní rozhodnutí bezpečnostnej politiky, sú pod rovnakou riadiacou kontrolou a pracujú pri obmedzeniach bezpečnostnej politiky, ktoré sú kompatibilné s bezpečnostnou politikou funkcie TOE.

– Predpokladá sa, že všetky vzdialené dôveryhodné systémy IT, ktorým dôveruje funkcia TSF a ktoré majú poskytovať údaje alebo služby TSF funkcii TOE alebo majú podporovať TSF v uplatňovaní rozhodnutí bezpečnostnej politiky, správne implementujú funkcie, ktoré používa TSF v súlade s predpokladmi, zadefinovanými pre túto funkciu.
– Zabezpečená je integrita nasledujúcich informácií:
  - Každý kód TSF, vrátane funkcie overovania integrity, ktorý sa zavádza a spúšťa pred spustením mechanizmu overovania integrity
  - Všetky údaje TSF, vrátane údajov TSF pre vykonávanie overovania integrity, ktoré používa kód TSF, zavádzaný a spúšťaný pred spustením mechanizmu overovania integrity
– Len VIOS: Každá úprava alebo poškodenie súborov funkcie TOE, uplatňujúcich bezpečnosť alebo podstatných pre bezpečnosť, ktorú spôsobil užívateľ alebo základná platforma úmyselne alebo náhodne, musí byť zistená administrátorom.

- Pripojiteľnosť: Všetky pripojenia k vzdialeným dôveryhodným systémom IT a zo vzdialených dôveryhodných systémov IT a medzi fyzicky oddelenými časťami TSF, ktoré nie sú chránené samotnou funkciou TSF, sú fyzicky alebo logicky chránené v prostredí funkcie TOE za účelom zabezpečenia integrity a utajenia prenášaných údajov a za účelom zabezpečenia autenticity koncových bodov komunikácie.

Získanie softvéru

Ak chcete získať softvér, postupujte takto:

1. Stiahnite si produkt.
2. Kliknite na tlačidlo Help v ponuke Entitled software support na ľavom paneli. Konfigurácia, vyhodnotená podľa všeobecných kritérií, vyžaduje, aby boli produkt a všetky aktualizácie získané na fyzickom médiu alebo použitím aplikácie Download Director.

Informácie o inštalácii produktu nájdete v dokumente Installing a BAS /EAL4+ system.

Inštalácia systému BAS/EAL4+:

Ak zvolíte túto voľbu, riadenie prístupu RBAC bude automaticky povolené.

Ak chcete počas inštalácie základného operačného systému nastaviť voľbu BAS/EAL4+, vykonajte nasledovné kroky:

1. Na obrazovke Installation and Settings vyberte položku **More Options**.
2. Pod More Options vyberte **Yes** pre voľbu BAS/EAL4+ a ak používate oddiel WPAR, pre voľbu TCB vyberte **No**. Ak používate prispôsobený súbor bosinst.data pre neobsluhovanú inštaláciu, môže byť voľba TCB nastavená na **Yes**.

Deaktivujte vzdialené prihlásenie ako užívateľ root pre inštaláciu BAS. Ak chcete deaktivovať vzdialené prihlásenie ako užívateľ root, po inštalácii spustite tento príkaz:

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

Do skupiny **SUADMIN** pridajte administračných užívateľov, aby sa mohli príkazom **su** prepnúť do režimu užívateľov root.

Voľba **Enable BAS and EAL4+ Technology** je k dispozícii len za týchto podmienok:

- Metóda inštalácie je nastavená na novú inštaláciu a úplne prepísanie.
- Vybratý jazyk je angličtina.
- 64-bitové jadro je povolené.
- Súborový systém JFS2 je povolený.

Keď je voľba **Enable BAS and EAL4+ Technology** nastavená na Yes, voľba **Trusted Computing Base** je tiež nastavená na Yes a jediné platné voľby pre **Desktop** sú NONE alebo CDE.

Ak vykonávate neobsluhovanú inštaláciu použitím prispôsobeného súboru bosinst.data, pole **INSTALL_TYPE** musíte nastaviť na CC_EVAL a nasledujúce polia musia byť nastavené takto:

```
control_flow:
 CONSOLE = ???
 PROMPT = yes
 INSTALL_TYPE = CC_EVAL
 INSTALL_METHOD = overwrite
 TCB = yes
 DESKTOP = NONE or 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 or C
 MESSAGES = en_US or C
```

Viac informácií o RBAC nájdete v časti Role Based Access Control (RBAC).

Prostredie NIM (Network Installation Management) pre BAS/EAL4+:

Inštalácia klientov technológie BAS/EAL4+ môže byť vykonaná použitím prostredia NIM (Network Installation Management).

Hlavný počítač NIM je nakonfigurovaný tak, aby poskytoval potrebné prostriedky na inštaláciu príslušnej úrovne BAS/EAL4+ systému AIX 7.1. Klienti NIM potom môžu byť nainštalovaný použitím prostriedkov umiestnených na počítači NIM master. Nastavením nasledujúcich polí v prostriedku **bosinst_data** môžete vykonať neobsluhovanú inštaláciu NIM klienta:

```
control_flow:
 CONSOLE = ???
 PROMPT = no
 INSTALL_TYPE = CC_EVAL
 INSTALL_METHOD = overwrite
 TCB = yes
 DESKTOP = NONE or 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 or C
 MESSAGES = en_US or C
```

Hlavný počítač NIM nemôže byť nakonfigurovaný ako systém BAS/EAL4+ a nemôže byť pripojený k rovnakej sieti s ostatnými systémami BAS/EAL4+. Keď spúšťate inštaláciu z hlavného počítača NIM, voľba ponuky **Remain NIM client after install SMIT** musí byť nastavená na hodnotu No. Po nainštalovaní klienta NIM ako systému BAS/EAL4+ musí byť klient NIM odstránený zo siete hlavného počítača NIM a ďalšie inštalácie a aktualizácie softvéru sa nemôžu vykonať pomocou hlavného počítača NIM.

Ukážkovou situáciou je mať dve sieťové prostredia; prvá sieť pozostáva z mastera NIM a systémov, ktoré nie sú systémami BAS/EAL4+; druhá sieť pozostáva len zo systémov BAS/EAL4+. Vykonajte inštaláciu NIM na klientovi NIM. Po dokončení inštalácie odpojte novonainštalovaný systém zo siete počítača BAS/EAL4+ NIM master a pripojte systém k vyhodnotenej sieti.

Druhý príklad pozostáva z jednej siete. Počítač NIM master nie je pripojený k sieti, keď iné systémy fungujú vo vyhodnotenej konfigurácii a systémy BAS/EAL4+ nie sú pripojené k sieti počas inštalácie NIM.

Softvérový balík BAS/EAL4+:

Keď vyberiete voľbu **BAS/EAL4+**, bude nainštalovaný obsah inštalačného balíka /usr/sys/inst.data/sys_bundles/CC_EVAL.BOS.autoi.

S vybratou voľbou **BAS/EAL4+** sa môžete voliteľne rozhodnúť pre inštaláciu balíka grafického softvéru a balíka softvéru dokumentačných služieb. Ak vyberiete voľbu **Graphics Software** s voľbou **BAS/EAL4+**, nainštalovaný bude obsah balíka softvéru /usr/sys/inst.data/sys_bundles/CC_EVAL.Graphics.bnd. Ak vyberiete voľbu Documentation Services Software s voľbou **BAS/EAL4+**, nainštalovaný bude obsah balíka softvéru /usr/sys/inst.data/sys_bundles/CC_EVAL.DocServices.bnd.

Po nainštalovaní produktov licenčných programov (LPP) systém zmení predvolenú konfiguráciu tak, aby vyhovovala požiadavkám BAS/EAL4+. V predvolenej konfigurácii sa vykonajú tieto zmeny:

- Odstránenie /dev/echo zo súboru /etc/pse.conf.
- Vytvorenie inštancií zariadení tokov.
- Povolenie prístupu k vymeniteľným médiám len užívateľovi s oprávneniami typu root.
- Odstránenie iných než CC položiek zo súboru inetd.conf.
- Zmena rôznych prístupových práv súborov.
- Registrácia symbolických prepojení v súbore sysck.cfg.
- Registrácia zariadení v súbore sysck.cfg.
- Nastavenie predvolených atribútov užívateľov a portov.
- Konfigurácia aplikácie doc_search na použitie prehliadača.
- Odstránenie httpdlite zo súboru inittab.
- Odstránenie súboru writesrv z inittab.
- Odstránenie mkatmpvc zo súboru inittab.
- Odstránenie atmsvcd zo súboru inittab.
- Vypnutie snmpd v súbore /etc/rc.tcpip.
- Vypnutie hostmibd v súbore /etc/rc.tcpip.
- Vypnutie snmpmibd v súbore /etc/rc.tcpip.
- Vypnutie aixmibd v súbore /etc/rc.tcpip.
- Vypnutie muxatmd v súbore /etc/rc.tcpip.
- Nastavenie portu NFS (2049) ako privilegovaný port.
- Pridanie chýbajúcich udalostí do súboru /etc/security/audit/events.
- Kontrola prevádzky rozhranie spätnej slučky.
- Vytvorenie synoným pre položku /dev/console.
- Vynútenie predvolených oprávnení pre pripojenie k serveru X-server.
- Zmena adresára /var/docsearch tak, aby boli všetky súbory globálne čitateľné.
- Pridanie odsekov pre Object Data Manager (ODM) na nastavenie oprávnení konzoly.
- Nastavenie oprávnení na pseudoterminálach v štýle BSD na hodnotu 000.
- Zakázanie súborov .netrc.
- Pridanie spracovania adresára opráv.

Grafické užívateľské rozhranie:

Systém vyhovujúci BAS/EAL4+ obsahuje systém X Windows ako grafické užívateľské rozhranie.

X Windows poskytuje mechanizmus pre zobrazenie grafických klientov, ako sú hodiny, kalkulačky a ďalšie grafické aplikácie, a rôznych terminálových relácií používajúcich príkaz **aixterm**. Systém X Windows sa spustí príkazom **xinit** z úvodného príkazového riadka po prihlásení sa užívateľa na hostiteľskú konzolu.

Ak chcete spustiť reláciu systému X Windows, zadajte:

```
xinit
```

Tento príkaz spustí X Windows server s mechanizmom lokálneho prístupu, povoleným iba pre vyvolávač. Klienti systému X Windows, ktorí majú UID nastavené na root, budú môcť pristupovať na X Windows server prostredníctvom soketu domény UNIX s použitím koreňového prepísania prístupových obmedzení. Klienti systému X Windows, ktorí majú UID nastavené na iných užívateľov alebo ktorých spustili iní užívatelia, nebudú môcť pristupovať na X Windows server. Toto obmedzenie zabráni iným užívateľom hostiteľa získať neautorizovaný prístup na server X Windows.

Inštalácia systému LAS/EAL4+:

Ak zvolíte túto voľbu, riadenie prístupu RBAC bude automaticky povolené.

Ak chcete počas inštalácie základného operačného systému nastaviť voľbu LAS/EAL4+, vykonajte nasledovné kroky:

Voľby inštalácie môžete upraviť zadaním 3, ak chcete zmeniť **bezpečnostný model**, a zadaním 4, ak chcete zobraziť **rozšírené voľby** v okne Installation and Settings. Tieto voľby sa menia v závislosti od typu inštalácie (overwrite, preservation alebo migration) a bezpečnostných volieb. Pre LAS je metóda inštalácie nová inštalácia alebo inštalácia úplným prepísaním. Vyberte **LAS/EAL4+ configuration install**.

Viac informácií o RBAC nájdete v časti Role Based Access Control (RBAC).

Inštalácia konfigurácie LAS/EAL4+ (k dispozícii len s dôveryhodným systémom AIX):

Voľba **LAS/EAL4+ configuration install** nainštaluje dôveryhodný systém AIX v režime nakonfigurovaného LAS/EAL4+. Režim nakonfigurovaného LAS/EAL4+ zabezpečuje v porovnaní s inštaláciou dôveryhodného systému AIX ďalšiu reštriktívnu bezpečnosť.

Ak vykonávate tichú inštaláciu pomocou prispôsobeného súboru bosinst.data, pole **INSTALL_TYPE** musí byť prázdne, pole **TRUSTED_AIX** by malo byť nastavené na yes a nasledujúce polia by mali byť nastavené na tieto hodnoty:

```
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 or C
 MESSAGES = en_US or C
```

Bližšie informácie o dôveryhodnom AIX nájdete v častiDôveryhodný AIX.

Prostredie NIM (Network Installation Management) pre LAS/EAL4+:

Inštalácia klientov technológie LAS/EAL4+ môže byť vykonaná použitím prostredia NIM (Network Installation Management).

Hlavný počítač NIM je nakonfigurovaný tak, aby poskytoval potrebné prostriedky na inštaláciu príslušnej úrovne LAS/EAL4+ systému AIX 7.1. Klienti NIM potom môžu byť nainštalovaný použitím prostriedkov umiestnených na počítači NIM master. Nastavením nasledujúcich polí v prostriedku bosinst_data môžete vykonať neobsluhovanú inštaláciu NIM klienta:

```
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 or C
 MESSAGES = en_US or C
```

Hlavný počítač NIM nemôže byť nakonfigurovaný ako systém LAS/EAL4+ a nemôže byť pripojený k rovnakej sieti s ostatnými systémami LAS/EAL4+. Keď spúšťate inštaláciu z hlavného počítača NIM, voľba ponuky **Remain NIM client after install SMIT** musí byť nastavená na hodnotu No. Po nainštalovaní klienta NIM ako systému LAS/EAL4+ musí byť klient NIM odstránený zo siete hlavného počítača NIM a ďalšie inštalácie a aktualizácie softvéru sa nemôžu vykonať pomocou hlavného počítača NIM.

Ukážkovou situáciou je mať dve sieťové prostredia; prvá sieť pozostáva z mastera NIM a systémov, ktoré nie sú systémami LAS/EAL4+; druhá sieť pozostáva len zo systémov LAS/EAL4+. Vykonajte inštaláciu NIM na klientovi NIM. Po dokončení inštalácie odpojte novonainštalovaný systém zo siete počítača LAS/EAL4+ NIM master a pripojte systém k vyhodnotenej sieti.

Druhý príklad pozostáva z jednej siete. Počítač NIM master nie je pripojený k sieti, keď iné systémy fungujú vo vyhodnotenej konfigurácii a systémy LAS/EAL4+ nie sú pripojené k sieti počas inštalácie NIM.

Fyzické prostredie systémov BAS/EAL4+ a LAS/EAL4+:

Systémy BAS/EAL4+ and LAS/EAL4+ majú špecifické požiadavky na prostredie, v ktorom sú spúšťané.

Požiadavky sú nasledovné:

- Fyzický prístup k systémom musí byť obmedzený tak, aby systémové konzoly mohli používať len oprávnení administrátori.
- Servisný procesor nie je pripojený k modemu.
- Fyzický prístup k terminálom je obmedzený na autorizovaných užívateľov.
- Fyzická sieť je zabezpečená voči odpočúvacím a falšovacím programom (nazývanými tiež trójske kone). Pri komunikácii cez nezabezpečené linky sú potrebné doplnkové opatrenia na zabezpečenie, napríklad šifrovanie.

- Komunikácia s inými systémami, ktoré nie sú systémami AIX 7.1 BAS/EAL4+ alebo LAS/EAL4+, alebo nie sú pod rovnakou riadiacou kontrolou, nie je povolená.
- Pri komunikácii s inými systémami BAS/EAL4+ a LAS/EAL4+ bude použitý len IPv4. IPv6 je súčasťou vyhodnocovanej konfigurácie, ale obsiahnuté sú iba funkčné danosti IPv6, ktoré podporuje aj IPv4.
- Užívatelia nesmú mať možnosť meniť systémový čas.
- Systémy v prostredí LPAR nemôžu zdieľať PHB.

Organizačné prostredie systémov BAS/EAL4+ a LAS/EAL4+:

Pre systémy BAS/EAL4+ a LAS/EAL4+ musia byť splnené určité procedurálne a organizačné požiadavky.

Potrebné je splnenie týchto požiadaviek:
- Administrátori musia byť dôveryhodní a riadne vyškolení.
- ID užívateľa sa v systéme prideľuje len tým užívateľom, ktorí majú oprávnenie pracovať s informáciami v systémoch.
- Užívatelia musia používať len kvalitné heslá (s náhodným obsahom, ktorý sa netýka užívateľa alebo organizácie). Informácie o pravidlách pre nastavenie hesiel nájdete v časti “Heslá” na strane 62.
- Užívatelia nesmú svoje heslá poskytnúť iným osobám.
- Administrátori musia mať dostatočné znalosti v oblasti riadenia systémov vyžadujúcich vysokú úroveň zabezpečenia.
- Administrátori musia pracovať v súlade s pravidlami obsiahnutými v dokumentácii k systému.
- Administrátori sa musia prihlásiť so svojim osobným ID a na prepnutie režimu superužívateľa na administráciu musia použiť príkaz **su**.
- Administrátormi generované heslá pre užívateľov systému musia byť daným užívateľom bezpečne prenesené.
- Osoby zodpovedné za systém musia vytvoriť a implementovať postupy nevyhnutné pre bezpečnú prevádzku systémov.
- Administrátori musia zabezpečiť, aby bol prístup k dôležitým bezpečnostným systémovým prostriedkom chránený nastavením bitov oprávnení a zoznamov ACL.
- Za účelom prenosu najcitlivejších informácií uložených v systémoch musí byť fyzická sieť schválená organizáciou.
- Procedúry údržby musia zahŕňať pravidelnú diagnostiku systémov.
- Administrátori musia mať na bezpečnú prevádzku a obnovu systému po havárii primerané procedúry.
- Premenná prostredia *LIBPATH* by sa nemala meniť, lebo by to mohlo spôsobiť, že dôveryhodný proces zavedie nedôveryhodnú knižnicu.
- V operačnom systéme sa nesmie použiť wiretapping a sledovanie softvéru (tcpdump, trace).
- Anonymné protokoly, ako je napríklad protokol HTTP, sa môžu používať len za účelom získavania verejných informácií (napr. on-line dokumentácia).
- Možno použiť len NFS založený na TCP.
- Užívateľom by sa nemal prideliť prístup k vymeniteľným médiám. Súbory zariadenia by mali byť chránené príslušnými zoznamami prístupových práv alebo bitmi oprávnení.
- Administrátori nesmú na alokovanie a dealokovanie prostriedkov používať dynamické oddiely. Konfigurácia oddielov môže byť vykonávaná len vtedy, ak nebežia vôbec žiadne oddiely.

Operačné prostredie systémov BAS/EAL4+ a LAS/EAL4+:

Pre systém BAS/EAL4+ a LAS/EAL4+ musia byť splnené určité operačné požiadavky a procedúry.

Potrebné je splnenie týchto požiadaviek a procedúr:
- Ak sa používa Hardware Management Console (HMC), HMC bude umiestnený vo fyzicky riadenom prostredí.
- Len autorizovaný personál môže pristupovať k pracovnému prostrediu a k HMC.
- Ak sa používa HMC, HMC možno použiť len pre nasledujúce úlohy:

  - Úvodná konfigurácia oddielov. Oddiel nemôže byť aktívny počas procesu konfigurácie.
  - Reštart "uviaznutých" oddielov
- HMC nesmie byť použitý cez operáciu nakonfigurovaného systému.
- Funkcia systému "volanie domov" musí byť zakázaná.
- Vzdialený modemový prístup k systému musí byť zakázaný.
- Ak je operačný systém AIX spustený v prostredí povoľujúcom LPAR, správca skontroluje v dokumentácii LPAR požiadavky na prevádzku logických oddielov EAL4+.
- Funkcia servisnej autority musí byť na logických oddieloch vypnutá.

Systémová konfigurácia BAS/EAL4+:

Môžete nakonfigurovať systém Base AIX Security (BAS) a Evaluation Assurance Level 4+ (EAL4+).

Skupiny **system**, **sys**, **adm**, **uucp**, **mail**, **security**, **cron**, **printq**, **audit** a **shutdown** sa považujú za administratívne skupiny. Do tejto skupiny pridávajte len dôveryhodných užívateľov.

Správa:

Administrátori sa musia prihlasovať použitím osobného konta užívateľa. Na administráciu systému ako užívateľ s oprávneniami typu root musia používať príkaz **su**.

Aby sa efektívne zabránilo uhádnutiu hesla konta typu root, povoľte len autorizovaným administrátorom používať príkaz **su** na konte typu root. Zvoľte tento postup:

1. Pridajte položku do stanzy **root** súboru /etc/security/user nasledovne:

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

2. Zadefinujte skupinu v súbore /etc/group, ktorá bude obsahovať ID užívateľov len autorizovaných administrátorov, nasledovným spôsobom:

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

Správcovia musia dodržiavať aj tieto procedúry:

- Vytvoriť a implementovať postupy, ktoré zabezpečia, že hardvérové, softvérové a firmvérové súčasti obsiahnuté v systéme budú distribuované, inštalované a konfigurované bezpečným spôsobom.
- Zabezpečiť, aby bol systém konfigurovaný tak, že len administrátor môže do daného systému zaviesť nový a dôveryhodný softvér.
- Implementáciou procedúr zabezpečiť, aby užívatelia vymazávali obrazovku pred odhlásením sa zo sériových prihlasovacích zariadení (napríklad terminálov 3151 spoločnosti IBM®).

Konfigurácia užívateľa a portu:

Voľby konfigurácie AIX pre užívateľov a porty musia byť nastavené na splnenie požiadaviek hodnotenia. Skutočná požiadavka je, aby TSF poskytovala mechanizmus správneho uhádnutia hesla, ktorý spĺňa kvalitu metriky. Pravdepodobnosť správneho uhádnutia hesla útočníkom počas životnosti hesla musí byť menšia ako 2^-20.

Súbor /etc/security/user, zobrazený v nasledujúcom príklade používa zoznam slovníka /usr/share/dict/words. Súbor /usr/share/dict/words je obsiahnutý v sade súborov bos.data. Je potrebné, aby ste nainštalovali sadu súborov bos.data skôr, ako začnete konfigurovať súbor /etc/security/user. Odporúčané hodnoty pre súbor /etc/security/user sú nasledovné:

```
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
```

Predvolené hodnoty v súbore /etc/security/user by nemali byť prepísané špecifickými nastaveniami pre jednotlivých užívateľov.

Poznámka: Nastavenie parametra login = false v sekcii root zabraňuje priamemu prihláseniu užívateľa s oprávneniami typu root. Prihlásiť sa ku kontu užívateľa s oprávneniami typu root môžu len užívatelia kont, ktoré disponujú privilégiami **su** pre konto užívateľa s oprávneniami typu root. Ak sa proti systému odosielajúcemu nesprávne heslá kontám užívateľov spustí útok typu Odmietnutie služby (DoS), daný útok môže spôsobiť zablokovanie všetkých kont užívateľov. Takýto útok môže všetkým užívateľom (vrátane užívateľov s oprávneniami administrátora) zabrániť v prihlásení do systému. Po zamknutí užívateľovho konta sa užívateľ nebude môcť prihlásiť, kým správca systému neresetuje užívateľov atribút unsuccessful_login_count v súbore /etc/security/lastlog tak, aby bol menší než hodnota užívateľského atribútu loginretries. Ak sú zablokované všetky administratívne kontá, bude pravdepodobne potrebné opätovne zaviesť systém do režimu údržby a spustiť príkaz **chsec**. Ďalšie informácie o používaní príkazu **chsec** nájdete v časti “Riadenie užívateľských kont” na strane 51.

Navrhované hodnoty pre súbor /etc/security/login.cfg sú nasledovné:

```
default:
  sak_enabled = false
  logintimes =
```

```
logindisable = 4
logininterval = 60
loginreenable = 30
logindelay = 5
```

Zoznam programov setuid/setgid:

Pre systémy AIX, podporujúce BAS, je vytvorený zoznam dôveryhodných aplikácií.

Bity **suid/sgid** sú vypnuté pre všetky nedôveryhodné programy, ktoré sú vo vlastníctve užívateľa s oprávneniami typu root alebo dôveryhodnej skupiny. Jediné programy v systéme po inštalácii BAS, ktoré sú buď **suid** a sú vo vlastníctve užívateľa s oprávneniami typu root alebo **sgid** a sú vo vlastníctve jednej z týchto dôveryhodných skupín sú **system**, **sys**, **adm**, **uucp**, **mail**, **security**, **cron**, **printq**, **audit** a **shutdown**. Do týchto skupín pridávajte len dôveryhodných užívateľov.

Zoznam dôveryhodných aplikácií je vytvorený posúdením všetkých aplikácií spadajúcich minimálne do jednej z nasledujúcich kategórií.

- Bit koreňa SUID pre príslušnú aplikáciu je zapnutý
- Bit SGID pre jednu z dôveryhodných skupín je zapnutý
- Aplikácie majúce prístup do niektorej z dôveryhodných databáz podľa návodu administrátora

Poznámka: Bit **setuid** pre príkaz **ipcs** by mal administrátor systému odstrániť. Administrátor systému by mal spustiť príkazy **chmod u-s /usr/bin/ipcs** a **chmod u-s /usr/bin/ipcs64**.

Zmena súborového systému auditu:

Ak zvolíte túto voľbu, riadenie prístupu RBAC bude automaticky povolené.

Súborový systém /audit je súborový systém jfs. Musí byť zmenený na súborový systém jfs2. Navyše, systémy BAS vyžadujú ďalšie príkazy. Ak chcete vykonať zmeny na súborovom systéme, postupujte takto:

1. Ak chcete zmeniť súborový systém pre systémy BAS, zadajte príkaz:

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

   Pre systémy LAS prejdite na krok 3.
2. Ak pole TYPE obsahuje symbol otáznika (?), zadajte príkaz:

   ```
   synclvodm -v rootvg
   ```
3. Odstráňte súborový systém jfs a vytvorte súborový systém jfs2 zadaním príkazu:

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

Aktualizácia databázy dôveryhodných podpisov (TSD):

Táto sekcia popisuje procedúru aktualizácie databázy TSD.

Konfigurácia BAS/LAS zmení bity režimu systému a nastanú chyby integrity TSD.

Počas opätovného zavedenia systému vyberte voľbu **Ignore All**.

Ak chcete zaktualizovať databázu TSD, zadajte príkaz:

```
trustchk -u ALL mode
```

Používanie systému LAS:

Táto sekcia poskytuje návod na používanie systému LAS.

Po nainštalovaní systému ako isso nastavte voľbu automatického opätovného zavedenia na **false** zadaním tohto príkazu:

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

Ak bude databáza TSD ďalej generovať chyby intlabel, vymažte ich použitím isso s privilégiom **PV_ROOT** zadaním týchto príkazov:

```
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
```

Ak sa na konzole zobrazia chybové správy, súvisiace s auditom, s privilégiom isso znova spustite systém auditu zadaním príkazov:

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

Po troch neúspešných pokusoch o prihlásenie je prihlásenie isso/so sieťou zablokované. Administrátor však môže ďalej pristupovať k týmto kontám na lokálnej konzole.

Výstup z príkazov, ktoré spustil cron/at, nebude postúpený do odkladacej oblasti pošty užívateľa.

Celosvetovo zapisovateľné adresáre s rozsahmi označenia (napríklad: /tmp) nie sú delené na oddiely. Za účelom zabránenia možnosti toku informácií medzi označeniami musí administrátor rozdeliť tieto adresáre okamžite po úvodnej konfigurácii.

Sieťové rozhranie:

Táto sekcia popisuje procedúru použitia sieťového rozhrania.

V dôveryhodnom systéme AIX má predvolené sieťové rozhranie rozsah označenia minSL=impl_lo a maxSL=ts_all. Pre systémy LAS/EAL4+ neexistuje rozsah označenia. Predvolené pravidlo sa automaticky zmení na impl_lo, keď vyberiete voľbu inštalácie LAS/EAL4+. Ak chcete zmeniť predvolené pravidlo ako isso, použite príkaz **netrule**.

Napríklad:

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

Aktualizácia oddielu WPAR:

Táto sekcia popisuje procedúru zmeny oddielov pracovného zaťaženia (WPAR) pre AIX na kompatibilné s EAL4+.

Vytvorte oddiel WPAR v systéme BAS a v tomto oddiele spustite nasledujúci príkaz, ktorý ho zmení na kompatibilný s EAL4+:

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

Keď spustíte clogin v systéme LAS po prvý raz, spustia sa skripty prvého zavedenia (medzi ktoré patrí CC_EVALify.sh).

Skripty prvého zavedenia majú za následok, že clogin beží dlhšie ako obvykle, keď clogin zavolá TSM za účelom prihlásenia. Oddiel WPAR je však stále v režime konfigurácie, takže prihlásenie je odmietnuté. Pred pokusom o ďalší

clogin musíte počkať približne 10 na dokončenie konfigurácie oddielu WPAR. Pre novo vytvorené systémy oddielov WPAR musia byť voľby predvoleného užívateľa nastavené tak, aby spĺňali požiadavky vyhodnotenia, medzi ktoré patria:

- root v režime BAS
- isso/sa/so v režime LAS

Užívatelia root a isso nemajú žiadne heslo alebo vyžadujú slabé heslo. Heslá musia byť aktualizované skôr, než povolíte nedôveryhodným užívateľom pristupovať do globálneho prostredia alebo príslušného oddielu WPAR.

Požiadavka na heslo pre vyhodnotenie je, že pravdepodobnosť správneho uhádnutia hesla musí byť najmenej jedno z 1.000.000 a pravdepodobnosť správneho uhádnutia hesla počas opakovaných pokusov v rámci jednej minúty musí byť najmenej jedno z 100.000. Aby sa vyhovelo tejto požiadavke, parametre užívateľa v súbore /etc/security/user sa zmenia na:

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

Aktualizácia EFS:

Táto sekcia popisuje procedúru nastavenia bezpečnostných atribútov EFS, ktorý bol vyhodnotený ako šifrovací súborový systém.

Vyhodnotenie nezahrňuje aspekty režimu ochrany koreňa pred úplným prístupom ku koreňu. Pri zapínaní EFS nastavte bezpečnostné atribúty pre príkazy **efsmgr** a **egskeymgr** spustením príkazu:

```
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
```

Vymazanie pevného disku:

AIX povoľuje vymazať pevné disky použitím servisného nástroja **Format media** v diagnostickom balíku systému AIX. Tento diagnostický balík je úplne zdokumentovaný v knihe *Diagnostic Information for Multiple Bus Systems* a v užívateľskej príručke pre váš hardvér.

Ak chcete vymazať pevný disk, spustite príkaz:

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

Tento príkaz spustí servisný nástroj **Format media** v rozhraní riadenom ponukou. Po výzve si vyberte svoj terminál.

Zobrazí sa zoznam pre výber prostriedkov. Z neho si vyberte zariadenia pevných diskov, ktoré chcete vymazať a potvrďte zmeny podľa pokynov na obrazovke.

Po potvrdení výberov vyberte z ponuky **Erase Disk** a potvrďte svoj výber. Vyberte **Yes**.

Ďalej uveďte, či chcete **Read data from drive** alebo **Write patterns to drive**. Vyberte **Write patterns to drive**.

Potom budete mať možnosť modifikovať voľby vymazania disku. Po zadaní preferovaných volieb vyberte **Commit Your Changes** . Disk je vymazaný.

Poznámka: Dokončenie tohto procesu môže trvať dlho.

Limity prostriedkov:

Pri nastavovaní limitov pre prostriedky v súbore /etc/security/limits sa uistite, že tieto limity vyhovujú požiadavkám procesov v systéme.

Konkrétne, veľkosť stack nikdy nenastavujte na hodnotu unlimited. Zásobník s neobmedzenou veľkosťou by mohol prepísať iné segmenty spusteného procesu. Veľkosť stack_hard musí byť tiež obmedzená.

Podsystém auditu:

Existuje niekoľko procedúr ochrany podsystému auditu.

- Nakonfigurujte podsystém auditu na zaznamenanie všetkých príslušných bezpečnostných aktivít užívateľov. Ak chcete zabezpečiť, aby bol diskový priestor potrebný pre auditovanie dostupný a nebol zmenšovaný inými spotrebiteľmi priestoru súborového systému, nastavte pre údaje auditu vyhradený súborový systém.
- Ochrana záznamov auditu (ako napríklad protokol auditu, súbory bin a všetky ostatné údaje uložené v adresári /audit) pred inými užívateľmi ako sú užívatelia s oprávneniami typu root.
- Pri používaní subsystému auditu musí byť pre systém BAS/EAL4+ nastavené auditovanie v režime **bin**. Informácie o nastavení subsystému auditu nájdete v časti “Nastavenie auditovania” na strane 139.
- Protokolu auditu by malo byť vyhradených najmenej 20% diskového priestoru v systéme.
- Ak je zapnutý audit, parameter *binmode* v stanze start v súbore /etc/security/audit/config by mal byť nastavený na panic. Parameter *freespace* v stanze bin by mal byť nakonfigurovaný na minimálnu hodnotu rovnajúcu sa 25 percentám diskového priestoru vyhradeného na ukladanie protokolov auditu. Parametre *bytethreshold* a *binsize* by mali byť nastavené na veľkosť 65 536 bajtov.
- Skopírujte záznamy auditu zo systému do trvalého úložného priestoru na archivovanie.

Nezdieľané súbory v distribuovanom systéme:

Nasledovné súbory v adresári /etc/security by nemali byť v distribuovanom systéme zdieľané - mali by zostať špecifické pre jednotlivé hostiteľské počítače:

/etc/security/failedlogin
Protokolový súbor pre zlyhané prihlásenia podľa hostiteľského počítača

/etc/security/lastlog
Informácie podľa užívateľa o poslednom úspešnom a neúspešnom prihlásení na tomto hostiteľskom počítači

/etc/security/login.cfg
Charakteristické prihlasovacie údaje špecifické pre hostiteľský počítač týkajúce sa dôveryhodných ciest, užívateľských prostredí prihlasovania a ďalších prihlasovacích informácií

/etc/security/portlog
Informácie podľa portu pre zablokované porty na tomto hostiteľskom počítači

Automaticky generované záložné súbory zdieľaných súborov takisto nie sú zdieľané. Názvy záložných súborov sú rovnaké ako názvy pôvodných súborov, no na začiatku obsahujú malé písmeno o.

Použitie funkcie DACinet pre riadenie prístupu do siete založené na užívateľovi a porte.:

Funkciu DACinet možno použiť na obmedzenie prístupu užívateľov k portom TCP.

Ďalšie informácie o funkcii DACinet nájdete v časti “Užívateľské riadenie prístupov na port TCP s Discretionary Access Control for Internet Ports (DACinet)” na strane 201. Napríklad, pri používaní samotnej funkcie DACinet na obmedzenie prístupu k portu TCP/25 v smere do počítača pre užívateľa s oprávneniami typu root budú môcť prístup k tomuto portu získať len užívatelia s oprávneniami typu root z hostiteľských počítačov kompatibilných so štandardom

BAS/EAL4+. Táto situácia zabraňuje možnosti napodobňovania e-mailov štandardnými užívateľmi pripojením sa k portu TCP/25 na napadnutom počítači prostredníctvom protokolu telnet.

Za účelom aktivácie zoznamov prístupových práv pre pripojenia TCP počas zavedenia systému sa z adresára /etc/inittab spúšťa skript /etc/rc.dacinet. Tento skript načíta definície v súbore /etc/security/acl a zavedie zoznamy prístupových práv do jadra. Porty, ktoré by nemali byť chránené zoznamami prístupových práv, by mali byť uvedené v súbore /etc/security/services, ktorý používa rovnaký formát ako súbor /etc/services.

Predpokladajme podsieť 10.1.1.0/24 pre všetky pripojené systémy, položky ACL na obmedzenie prístupu len pre užívateľa s oprávneniami typu root pre X (TCP/6000) v súbore /etc/security/acl by boli nasledovné:

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

Inštalácia ďalšieho softvéru do systému kompatibilného so štandardom BAS/EAL4+:

Administrátor môže v systéme kompatibilnom so štandardom BAS/EAL4+ nainštalovať ďalší softvér. Ak softvér nie je spustený užívateľom s oprávneniami typu root alebo s privilégiami užívateľa s oprávneniami typu root, zruší sa platnosť kompatibility a BAS/EAL4+. Typickým príkladom sú kancelárske aplikácie, ktoré sú spúšťané bežnými užívateľmi a neobsahujú žiadne súčasti SUID.

Navyše zruší nainštalovaný softvér spustený s privilégiami užívateľa s oprávneniami typu root platnosť s BAS/EAL4+. Znamená to napríklad, že by sa nemala vykonávať inštalácia ovládačov pre starší súborový systém JFS, pretože tieto ovládače sa spúšťajú v režime jadra. Žiadne aplikácie, ktoré majú udelené jedno alebo viac privilégií prostredníctvom /etc/security/privcmds, nie sú akceptovateľné. Ďalší démoni, ktorí sa spúšťajú s privilégiami užívateľa s oprávneniami typu root (napríklad, démon SNMP), takisto spôsobujú neplatnosť kompatibility so štandardom BAS/EAL4+. Zapnutý systém BAS/EAL4+nemožno aktualizovať (bežne).

Systém kompatibilný so štandardom BAS/EAL4+ sa len zriedkavo používa v hodnotenej konfigurácii, obzvlášť v priemyselnom prostredí. Na to, aby bol systém produkcie založený na hodnotenom systéme, no nemusel byť pritom kompatibilný s presnými špecifikáciami daného hodnoteného systému, sú obyčajne potrebné ďalšie služby.

Zoznam prístupových práv v NSF v4 a politika obsahu:

Zoznam prístupových práv (ACL) v NFS v4 obsahuje polia **Type**, **Mask** a **Flags**.

Nasleduje popis týchto polí:

- Pole **Type** obsahuje jednu z nasledujúcich hodnôt:
  - ALLOW – Poskytuje subjektu, uvedenému v poli **Who**, oprávnenia špecifikované v poli **Mask**.
  - DENY – Zamieta subjektu, uvedenému v poli **Who**, oprávnenia špecifikované v poli **Mask**.
- Pole Mask obsahuje jednu alebo viac nasledujúcich hodnôt s jemným rozlíšením oprávnení:
  - READ_DATA / LIST_DIRECTORY – Čítať údaje z objektu, ktorý nie je adresárom, alebo zoznam objektov v adresári.
  - WRITE_DATA / ADD_FILE – Zapisovať údaje do objektu, ktorý nie je adresárom, alebo pridať neadresárový objekt do adresára.
  - APPEND_DATA / ADD_SUBDIRECTORY – Pridať údaje do objektu, ktorý nie je adresárom, alebo pridať podadresár do adresára.
  - READ_NAMED_ATTRS – Čítať pomenované atribúty objektu.
  - WRITE_NAMED_ATTRS – Zapisovať pomenované atribúty objektu.
  - EXECUTE – Spustiť súbor alebo prechádzať/prehľadávať adresár.
  - DELETE_CHILD – Vymazať súbor alebo adresár z adresára.
  - READ_ATTRIBUTES – Čítať základné (nie ACL) atribúty súboru.
  - WRITE_ATTRIBUTES – Zmeniť časy asociované so súborom alebo adresárom.
  - DELETE – Vymazať súbor alebo adresár.

  - READ_ACL – Čítať zoznam ACL.
  - WRITE_ACL – Zapísať zoznam ACL.
  - WRITE_OWNER – Zmeniť vlastníka a skupinu.
  - SYNCHRONIZE – Synchronizovať prístup (existuje pre kompatibilitu s inými klientmi NFS v4, ale nemá žiadnu implementovanú funkciu).
- **Flags** – Toto pole definuje dedičné danosti ACL zoznamov adresára a indikuje, či pole **Who** obsahuje skupinu alebo nie. Toto pole obsahuje nula alebo viac nasledujúcich príznakov:
  - **FILE_INHERIT** – Špecifikuje, že v tomto adresári budú novo vytvorené položky, ktoré nie sú adresárom, dediť túto položku.
  - **DIRECTORY_INHERIT** – Špecifikuje, že v tomto adresári budú novo vytvorené podadresáre dediť túto položku.
  - **NO_PROPAGATE_INHERIT** – Špecifikuje, že v tomto adresári budú novo vytvorené podadresáre dediť túto položku, ale takéto novo vytvorené podadresáre danú položku už neodovzdajú podadresárom v nich vytvoreným.
  - **INHERIT_ONLY** – Špecifikuje, že táto položka sa nevzťahuje na tento adresár, iba na novo vytvorené objekty, ktoré dedia túto položku.
  - **IDENTIFIER_GROUP** – Špecifikuje, že pole **Who** reprezentuje skupinu; inak pole **Who** reprezentuje užívateľa alebo špeciálnu hodnotu **Who**.
- **Who** - Toto pole obsahuje jednu z nasledujúcich hodnôt:
  - User – Špecifikuje užívateľa, na ktorého sa vzťahuje táto položka.
  - Group – Špecifikuje skupinu, na ktorú sa vzťahuje táto položka.
  - Special – Tento atribút môže nadobudnúť jednu z nasledujúcich hodnôt:
    - OWNER@ – Určuje, že táto položka sa vzťahuje na vlastníka objektu.
    - GROUP@ – Určuje, že táto položka sa vzťahuje na skupinu vlastniacu objekt.
    - EVERYONE@ – Určuje, že táto položka sa vzťahuje na všetkých užívateľov systému vrátane vlastníka a skupiny.

Ak je ACL prázdny, k objektu môže pristúpiť iba subjekt s účinným UID = 0. Vlastník objektu má implicitne nasledujúce hodnoty masky, bez ohľadu na to, čo zoznam ACL obsahuje alebo neobsahuje:

- READ_ACL
- WRITE_ACL
- READ_ATTRIBUTES
- WRITE_ATTRIBUTES

Hodnota APPEND_DATA je implementovaná ako WRITE_DATA. Pokiaľ ide o účinok, medzi hodnotou WRITE_DATA a APPEND_DATA nie je funkčný rozdiel. Obe hodnoty musia byť nastavené alebo musia mať zrušené nastavenie vo vzájomnej zhode.

Vlastníctvo objektu sa dá modifikovať použitím hodnoty WRITE_OWNER. Keď sa zmení vlastník alebo skupina, bit **setuid** sa vypne. Príznaky dedičnosti majú význam iba v ACL adresára a platia iba pre objekty, ktoré boli vytvorené v adresári po nastavení príznakov dedičnosti (napríklad existujúce objekty nebudú ovplyvnené zmenami dedičnosti v ACL adresára). Položky v ACL NFS v4 sú závislé od poradia. Aby sa zistilo, či je povolený požadovaný prístup, každá položka sa spracuje v príslušnom poradí. Do úvahy sa berú iba položky, ktoré majú nasledujúce hodnoty:

- Pole **Who** zhodné s účinným UID
- Užívateľ zadaný v položke alebo účinné GID
- Skupina zadaná v položke subjektu

Každá položka sa spracúva, kým nedostanú povolenie všetky bity prístupu žiadateľa. Keď položka povolí (ALLOW) typ prístupu, už sa viac neberie do úvahy pri spracúvaní ďalších položiek. Ak sa vyskytne položka s odmietnutím (DENY), pričom pre danú hodnotu masky je potrebný prístup žiadateľa a nie je určený, takáto požiadavka bude zamietnutá. Ak hodnotenie dosiahne koniec zoznamu ACL, požiadavka sa zamietne.

Maximálna podporovaná veľkosť ACL je 64 KB. Každá položka v ACL môže mať inú dĺžku, a 64 KB je jediné obmedzenie, ktoré pre ňu platí.

Hodnota WRITE OWNER:

Politika v NFS v4 poskytuje kontrolu nad tým, kto môže čítať a zapisovať atribúty objektu.

Subjekt s účinným UID 0 môže vždy obísť politiku NFS v4. Vlastník objektu môže povoliť iným čítať a zapisovať atribúty objektu použitím atribútov masky ACL READ_ATTRIBUTES, WRITE_ATTRIBUTES, READ_NAMED_ATTRS a WRITE_NAME_ATTRS. Vlastník môže určovať, kto môže čítať a zapisovať ACL pomocou hodnôt masky ACL READ_ACL a WRITE_ACL. Vlastník objektu má vždy prístup READ_ATTRIBUTES, WRITE_ATTRIBUTES, READ_ACL a WRITE_ACL. Vlastník objektu môže tiež iným povoliť zmeniť vlastníka a skupinu objektu pomocou atribútu WRITE_OWNER. Vlastník objektu nemôže štandardne meniť vlastníka alebo skupinu objektu, môže však pridať položku WRITE_OWNER do ACL, v ktorej špecifikuje sám seba, alebo môže objekt zdediť položku ACL, ktorá špecifikuje položku WRITE_OWNER s hodnotou pre **Who** OWNER@. Keď sa zmení vlastník alebo skupina, bit **setuid** sa vypne.

Pre tieto pravidlá platia určité výnimky:

- Ak je vlastníkom objektu UID 0, iba UID 0 môže zmeniť vlastníka, avšak skupinu ďalej bude môcť zmeniť subjekt s atribútom WRITE_OWNER.
- Za predpokladu, že objekt má pre určitý subjekt atribút WRITE_OWNER, vo verziách systému AIX 5.3 spred technologickej úrovne 5300-05 platí, že ak má objekt vlastníka bez UID 0, takýto vlastník môže byť zmenený zase len na užívateľa bez UID 0. V systéme AIX s 5300-05 a neskorších verziách platí, že ak má objekt vlastníka bez UID 0, takýto vlastník môže byť zmenený iba na EUID subjektu, ktorý sa pokúša zmeniť vlastníka.
- Skupinu je možné zmeniť na ľubovoľnú skupinu z množiny súbežne sa vyskytujúcich skupín subjektu, s výnimkou, že nikdy nemôže byť zmenená na GID 0 alebo GID 7 (systémová alebo bezpečnostná), a to ani vtedy, keď tieto skupiny patria do množiny súbežne sa vyskytujúcich skupín subjektu.

Podporované LDAP a súborové administračné databázy:

Hodnotenie nepodporuje administračné databázy NFS. Autentifikačné metódy typu DCE a NIS nie sú podporované.

Hodnotenie podporuje iba:

- Autentifikácia založená na súboroch (predvolené)
- Autentifikácia LDAP systému UNIX (použite server LDAP IBM Tivoli Directory Server v 6.0)

Bližšie informácie o autentifikácii založenej na súboroch nájdete v téme Autentifikácia užívateľa.

Autentifikácia cez LDAP:

I&A založená na LDAP sa konfiguruje v režime autentifikácie "UNIX-type". V tomto režime sú administračné údaje (vrátane mien, ID a hesiel užívateľom) uložené v LDAP, kde sa prístup k údajom obmedzuje na administrátora LDAP.

Keď sa užívateľ prihlási na systém, systém sa naviaže na LDAP server prostredníctvom konta administrátora LDAP cez SSL pripojenie, získa potrebné údaje pre užívateľa (vrátane hesla) z LDAP, a potom vykoná autentifikáciu s použitím údajov, ktoré získal z LDAP. Systém administračnú databázu uchováva na LDAP serveri. Zvyšní hostitelia importujú administračné údaje z toho istého LDAP servera pomocou rovnakého mechanizmu, ako bolo opísané vyššie. Systém zachováva administračnú databázu konzistentnou tým, že všetky administračné zmeny vykonáva na určenom LDAP serveri. ID užívateľa na niektorom počítači označuje rovnakú osobu aj na všetkých ostatných počítačoch. Navyše, konfigurácia hesla, mapovania mien na UID a ďalšie údaje sú na všetkých hostiteľoch v distribuovanom systéme identické.

Bližšie informácie o nastavení autentifikácie s použitím LDAP si prečítajte v téme Light Directory Access Protocol. Bližšie informácie o nastavení SSL na LDAP si prečítajte v témach Nastavenie SSL na serveri LDAP a Nastavenie SSL na klientovi LDAP.

server LDAP:

Príkaz **mksecldap -s** nastaví systém AIX ako LDAP server na účely overenia bezpečnosti a správy údajov.

Vykonajte nasledujúce úlohy:

- Použite schému RFC2307AIX s voľbou **-S**.
- Pomocou voľby **-k** nakonfigurujte server tak, aby používal protokol SSL (Secure Sockets Layer). Táto akcia vyžaduje inštaláciu sady súborov **GSKit V8** a sady súborov **idsldap.clt_max_crypto32bit63.rte** pre 32-bitové systémy alebo sady súborov **idsldap.clt_max_crypto64bit63.rte** pre 64-bitové systémy. Pomocou pomocného programu ikeyman vygenerujte dvojice kľúčov pre adresárový server.

Užívateľské voľby LDAP musia byť nastavené, aby boli splnené požiadavky hodnotenia. Schéma RFC2370AIX definuje užívateľské atribúty. Použite rovnaké hodnoty, ako je popísané v téme BAS/EAL4+ system configuration. Administrátori servera Tivoli Directory Server nemusia pravidelne meniť svoje heslá (napríklad, pre heslá administrátorov neexistuje hodnota **MaxAge**). Kvôli tomu je treba meniť administračné heslá pre LDAP tak často, ako v prípade užívateľa AIX (**MaxAge** = 8 (týždňov)).

Vo vydaní Tivoli Directory Server 6.3 sa spracovanie zlyhaní autentifikácie nevzťahuje na administrátora servera ani na členov skupiny administrátorov. Na kontá administrátorov sa nevzťahujú ani pravidlá na vytvorenie hesla. Tieto pravidlá sa musia dodržiavať, ak sa používa Tivoli Directory Server 6.3.

Ak administrátor nepoužíva spoločnú koncovú databázu LDAP na správu užívateľov, musí zabezpečiť, aby bola databáza obsahujúca prihlasovacie údaje užívateľov konzistentne udržiavaná na jednotlivých systémoch TOE (TCP Offload Engine), ktoré sú súčasťou jednej siete. Príklady sú:

- /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

Súvisiace informácie:

Informácie o balíkoch, sadách súborov a požiadavkách IBM Tivoli Directory Server

Klient LDAP:

Príkaz **mksecldap -c** nastaví systém AIX ako klienta LDAP na účely overenia bezpečnosti a správy údajov.

Vykonajte nasledujúce úlohy:

- Použite príkaz **mksecldap -c** a zadajte **unix_auth** pre **authType** s voľbou **-A**.
- Nastavte klienta na použitie SSL použitím voľby **-k** v príkaze **mksecldap -c**. Zadanie klientskeho SSL kľúča vyžaduje inštalovanie sady súborov **GSKit** a **ldap.max_crypto_client**. Pomocou pomocného programu **gsk7ikm** vygenerujte páry kľúčov pre adresárový server.

NFS v4 Client/Server a Kerberos:

Prostredie NFS v4 Client/Server obsahuje LDAP na uchovávanie autentifikačných údajov a Kerberos na vytvorenie dôveryhodného kanála medzi klientmi a servermi NFS v4. Vyhodnocovaná konfigurácia podporuje službu NAS v1.4 pre Kerberos a IBM Tivoli Directory Server v6.0 (server LDAP) pre databázu užívateľov.

NAS v1.4 (server Kerberos verzia 5) musí byť nakonfigurovaný na použitie LDAP pre svoje databázy. Lístky Kerberos, ktoré predtým vydal server Kerberos, sú platné až do ich exspirácie.

Keď používate autentifikáciu cez Kerberos, splnomocnenie použité vo vzdialených volaniach procedúry, ktoré inicioval užívateľ, sú priradené k aktuálnemu lístku Kerberos, ktorého držiteľom je užívateľ, a nie je ovplyvnené reálnym ani účinným UID procesu. Keď k vzdialenému súborovému systému NFS pristupujete s použitím autentifikácie cez Kerberos počas spúšťania programu **setuid**, UID, ktoré vidí server, je založené na identite Kerberos, a nie UID, ktoré vlastní spúšťaný program **setuid**.

Hodnotená konfigurácia zahŕňa nastavenie NFS na použitie bezpečnosti RPCSEC-GSS. Bližšie informácie si pozrite v témach Network File System, Konfigurácia NFS servera a Konfigurácia klienta NFS. Pri nastavovaní servera zvoľte autentifikáciu cez Kerberos a na serveri povoľte rozšírenú bezpečnosť. Povolíte ju prostredníctvom SMIT s použitím príkazu **chnfs**. Príkaz **chnfs** má voľbu na povolenie bezpečnosti RPCSEC_GSS. Keď nastavujete klienta, postupujte podľa pokynov na použitie protokolu Kerberos, uvedených v téme Konfigurácia klienta NFS. Pozrite si tému Nastavenie siete pre RPCSEC-GSS, kde nájdete pokyny na nastavenie údajového servera so šifrovaním DES3 kvôli zabezpečeniu. Hodnotená konfigurácia podporuje iba šifrovanie des3.

Pravidlá pre heslá:

Vyhodnocovaná konfigurácia by mala mať tieto hodnoty pre pravidlá týkajúce sa hesiel, pokiaľ používate server Kerberos s LDAP ako databázou.

Viac informácií o pravidlách pre heslá si prečítajte v kapitole 9 "Managing Network Authentication Service passwords" v príručke *IBM Network Authentication Service Version 1.4 for AIX, Linux and Solaris Administrator's and User's Guide*.

Nasleduje zoznam hodnôt:

mindiff
4

maxrepeats
2

minalpha
2

minother
2

minlen 8

minage
0

histsize
10

Ak chcete, aby klient AIX NFS v4 a server AIX NFS v4 vzájomne bezpečne komunikovali a explicitne používali iba typy kódovania DES3, vytvorte principál servera "nfs/hostname" s DES3 enctype (napríklad des3-cbc-sha1), spolu so zodpovedajúcou položkou v súbore keytab (s použitím rozhrania **kadmin**), a urobte z DES3 (napríklad **des3-cbc-sha1**) prvú položku v časti **default_tgs_enctypes** súboru /etc/krb5/krb5.conf na klientskom počítači NFS v4.

Virtuálny I/O server:

Virtuálny I/O server (VIOS) je trvalo umiestnený na inom oddiele LPAR a poskytuje základné voliteľné riadenie prístupu medzi ovládačmi zariadení VIOS SCSI, operujúcimi v mene LPAR oddielov, a SCSI logickými jednotkami a fyzickými jednotkami prostredníctvom mapovaní.

LPAR oddiel (prostredníctvom ovládača zariadenia VIOS SCSI) môže byť mapovaný na 0 alebo viac logických a fyzických jednotiek, ale jednotka môže byť namapovaná len na jeden LPAR oddiel. Toto mapovanie ohraničuje LPAR oddiel iba na jednotky, ktoré sú k nemu priradené. VIOS tiež riadi mapovanie ovládačov zariadení ethernetového adaptéra VIOS na ovládače ethernetových zariadení VIOS, ktoré operujú v mene skupín oddielov LPAR, zdieľajúcich virtuálnu sieť. Vo vyhodnocovanej konfigurácii je povolené iba mapovanie (jeden-k-jednému) ovládača zariadení ethernetového adaptéra na ovládač ethernetového zariadenia, operujúceho v mene skupiny LPAR oddielov. Toto mapovanie typu jeden k jednému konfiguruje administrátor a vynucujú ho ovládače zariadenia. Ethernetové pakety ďalej nesmú byť vo vyhodnocovanej konfigurácii označené značkou VLAN. Pomocou tohto mechanizmu sa dá ohraničiť, ktoré oddiely LPAR uvidia určité ethernetové pakety.

Rozhranie VIOS by malo byť chránené pred prístupom neprivilegovaných užívateľov. Voľby užívateľa VIOS musia byť nastavené tak, aby vyhoveli požiadavkám vyhodnotenia. Skutočná požiadavka je, aby TSF poskytovala mechanizmus na overenie, či tajné informácie spĺňajú nasledujúcu kvalitu metriky: Pravdepodobnosť získania tajných informácií útočníkom počas životnosti tajných informácií je menšia ako 2^-20. Nasledujúce parametre by mali byť zmenené pre užívateľa v adresári /etc/security/user:

maxage
8

maxexpired
1

minother
2

minlen 8

maxrepeats
2

loginretries
3

histexpire
52

histsize
20

Ak chcete zmeniť predvolené nastavenia, použite tieto príkazy:

```
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
```

Keď hlavný administrátor (**padmin**) vytvorí nového užívateľa, je preňho potrebné explicitne zadať užívateľské atribúty. Napríklad na vytvorenie užívateľa s menom *davis* použije hlavný administrátor **padmin** nasledujúci príkaz:

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

Hlavný administrátor **padmin** musí tiež zastaviť nasledujúce démony a potom systém opätovne zaviesť:

- Odstránenie **writesrv** a **ctrmc** zo súboru /etc/inittab:

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

- Aby ste zabránili spusteniu démona v čase zavedenia, odstráňte súbory /etc/rc.d/rc2.d/Ksshd a /etc/rc.d/rc2.d/Ssshd. Po opätovnom zavedení zastavte démonov RSCT:

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

Všetci užívatelia, bez ohľadu na ich roly, by mali byť braní ako užívatelia s administrátorskými právami.

Administrátor systému môže spustiť všetky príkazy okrem príkazov v nasledujúcom zozname, ktoré sú vyhradené len pre hlavného administrátora (**padmin**):

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

Riadenie prihlásenia

Po nainštalovaní systému môžete z bezpečnostných dôvodov zmeniť predvolené hodnoty na prihlasovacej obrazovke.

Potenciálny hackeri dokážu zo štandardnej prihlasovacej obrazovky systému AIX získať cenné informácie, napríklad názov hostiteľa a verziu operačného systému. Tieto informácie by im mohli pomôcť pri pokuse zneužiť údaje systému. Z bezpečnostných dôvodov budete možno chcieť zmeniť predvolené nastavenia prihlasovacej obrazovky čo najskôr po inštalácii systému.

Podobné bezpečnostné otázky sa týkajú aj pracovných plôch KDE a GNOME. Viac informácií o systémoch KDE a GNOME nájdete v príručke *Installation and migration*.

Informácie o užívateľoch, skupinách a heslách nájdete v časti “Užívatelia, skupiny a heslá” na strane 46.

Nastavenie ovládacích prvkov prihlásenia:

Ovládacie prvky prihlásenia môžete nastaviť v súbore /etc/security/login.cfg.

Pre sťaženie útoku na systém pomocou uhádnutia hesla nastavte riadenia prihlásenia v súbore /etc/security/login.cfg nasledovným spôsobom:

Tabuľka 1. Atribúty a odporúčané hodnoty pre riadenie prihlasovania

Atribút	Platí pre pseudoterminály PtY (Sieť)	Platí pre terminály TTY	Odporúčaná hodnota	Poznámky
sak_enabled	Y	Y	false	Funkcia Secure Attention key sa vyžaduje len zriedkavo. Pozrite si časť "Používanie Secure Attention Key" na strane 5.
logintimes	N	Y		Tu zadajte povolené časové limity pre prihlásenie.
logindisable	N	Y	4	Prihlásenie na tomto termináli sa zakáže po 4 za sebou nasledujúcich neúspešných pokusoch.
logininterval	N	Y	60	Terminál sa vypne, ak zadaný počet neplatných pokusov o prihlásenie nastane v priebehu 60 sekúnd.
loginreenable	N	Y	30	Aktivujte terminál po tom, ako bol automaticky deaktivovaný po 30 minútach.
logindelay	Y	Y	5	Doba v sekundách medzi jednotlivými výzvami na prihlásenie. Vynásobí s číslom neúspešných pokusov. Napríklad 5, 10, 15, 20 sekúnd pri počiatočnej hodnote 5.

Tieto obmedzenia portov fungujú väčšinou na pripojených sériových termináloch, nie na pseudoterminálach, používaných prihláseniami do siete. V súbore môžete priamo špecifikovať terminály, napríklad:

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

Zmena uvítacej správy na prihlasovacej obrazovke:

Aby sa zabránilo zobrazovaniu určitých informácií na prihlasovacích obrazovkách, upravte parameter *herald* v súbore /etc/security/login.cfg.

Parameter *herald* štandardne obsahuje uvítaciu správu, v ktorej sa zobrazuje výzva na prihlásenie. Na zmenu tohto parametra použite príkaz **chsec** alebo súbor upravte priamo.

Nasledujúci príklad používa príkaz **chsec** na zmenu štandardného parametra *herald*:

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

Viac informácií o príkaze **chsec** nájdete v príručke *Commands Reference, Volume 1*.

Ak chcete upraviť tento súbor priamo, otvorte súbor /etc/security/login.cfg a aktualizujte parameter *herald* nasledovne:

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

Poznámka: Aby bol systém bezpečnejší, nastavte premenné *logindisable* a *logindelay* na číslo väčšie než 0 (# > 0).

Zmena prihlasovacej obrazovky pre CDE (common desktop environment):

Otázky bezpečnosti sa týkajú aj užívateľov, ktorí pracujú v prostredí Spoločné prostredie pracovnej plochy (CDE). Na prihlasovacej obrazovke CDE sa tiež štandardne zobrazuje názov hostiteľa a verzia operačného systému. Aby sa zabránilo zobrazovaniu týchto informácií, upravte súbor /usr/dt/config/$LANG/Xresources, kde **$LANG** odkazuje na lokálny jazyk nainštalovaný na vašom počítači.

V našom príklade, za predpokladu, že **$LANG** je nastavené na **C**, skopírujte tento súbor do adresára /etc/dt/config/C/Xresources. Potom otvorte súbor /usr/dt/config/C/Xresources a odstráňte v ňom uvítanie, ktoré obsahuje názov hostiteľa a verziu operačného systému.

Viac informácií o otázkach zabezpečenia prostredia CDE nájdete v “Zvládnutie starostí s X11 a CDE” na strane 39.

Zakázanie zobrazenia mena užívateľa a zmena výzvy na zadanie hesla:

V zabezpečenom prostredí môže byť nutné skryť zobrazovanie prihlasovacieho mena užívateľa, alebo poskytnúť prispôsobenú výzvu na zadanie hesla, ktorá sa odlišuje od štandardnej.

Štandardné správanie správ v dialógoch pri zadávaní mena a hesla vyzerá takto:

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

Ak si želáte zamedziť zobrazovanie mena používateľa v dialógoch a chybových hláseniach systému, upravte parameter *usernameecho* v súbore /etc/security/login.cfg. Predvolená hodnota pre *usernameecho* je true, čo znamená, že meno používateľa sa bude zobrazovať. Na zmenu tohto parametra použite príkaz **chsec** alebo súbor upravte priamo.

Nasledujúci príklad používa príkaz **chsec** na zmenu štandardného parametra *usernameecho* na false:

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

Viac informácií o príkaze **chsec** nájdete v príručke *Commands Reference, Volume 1*.

Ak chcete súbor upravovať priamo, otvorte súbor /etc/security/login.cfg a doplňte alebo upravte parameter *usernameecho* nasledovným spôsobom:

```
default:
 usernamecho = false
```

Nastavenie parametra *usernameecho* na hodnotu false spôsobí, že meno používateľa sa nebude v prihlasovacom dialógu zobrazovať. Na mieste mena sa budú v systémových dialógoch a chybových hláseniach zobrazovať znaky '*'. Budú vyzerať takto:

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

Dialóg zadávania hesla možno upraviť samostatne tak, aby sa v ňom zobrazoval zvolený reťazec. Je treba nastaviť parameter *pwdprompt* v súbore /etc/security/login.cfg. Predvolenou hodnotou tu je reťazec "*user*'s Password: ", kde *user* používateľ sa nahradí menom autentifikovaného používateľa.

Na zmenu tohto parametra použite príkaz **chsec** alebo súbor upravte priamo.

Nasledujúci príklad používa príkaz **chsec** na zmenu štandardného parametra *pwdprompt* na "Password: ":

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

Ak chcete súbor upravovať priamo, otvorte súbor /etc/security/login.cfg a doplňte alebo upravte parameter *pwdprompt* nasledovným spôsobom:

```
default:
 pwdprompt = "Password: "
```

Nastavenie parametra *pwdprompt* hodnotu "Password: " bude mať za následok, že takto predpísaný dialóg sa bude zobrazovať pri prihlasovaní ako aj v iných aplikáciách, ktoré používajú systémový dialóg na zadávanie hesla. Po nakonfigurovaní prispôsobeného dialógu sa bude tento dialóg pri prihlasovaní správať takto:

```
login: foo
Password:
```

Nastavenie predvolených systémových parametrov prihlasovania:

Ak chcete nastaviť predvolené systémové parametre prihlasovania, upravte súbor /etc/security/login.cfg.

Ak chcete nastaviť základné predvolené hodnoty pre viaceré parametre prihlasovania, napríklad tie, ktoré by ste mohli nastaviť pre nového užívateľa (počet pokusov o prihlásenie, opätovné prihlasovanie a interval prihlasovania), upravte súbor /etc/security/login.cfg .

Zabezpečenie terminálov bez obsluhy:

Ak chcete zabezpečiť svoj terminál, použite príkazy **lock** a **xlock**.

Každý systém je zraniteľný, ak sú terminály prihlásené a pritom ponechané bez obsluhy. Najväčší problém nastane, keď správca systému odíde od terminálu, ktorý bol zapnutý s oprávnením typu root. Platí, že užívatelia by sa mali odhlásiť vždy, keď odchádzajú od terminálu. Systémové terminály ponechané bez zabezpečenia znamenajú možné bezpečnostné riziko. Na zamknutie terminálu použite príkaz **lock**. Ak pracujete v rozhraní AIXwindows, použite príkaz **xlock**.

Povolenie automatického odhlásenia:

Zapnutím automatického odhlásenia zabránite útočníkovi, aby ohrozoval bezpečnosť systému.

Ďalší bezpečnostný problém vyplýva z toho, že užívatelia aj počas dlhšej neprítomnosti ponechávajú prístup ku svojim kontám. Cudzie osoby tak majú možnosť prevziať ovládanie terminálu užívateľa, čo môže ohroziť bezpečnosť systému.

Aby sa zabránilo tomuto možnému ohrozeniu bezpečnosti, môžete aktivovať automatické odhlásenie zo systému. Ak tak chcete spraviť, premenné prostredia TMOUT a TIMEOUT nastavte na počet sekúnd nečinnosti. Po uplynutí doby nečinnosti budete automaticky odhlásený, ako v nasledujúcom príklade:

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

V uvedenom príklade číslo 600 predstavuje počet sekúnd rovnajúci sa 10 minútam. Táto metóda pracuje iba z aplikácie prostredia shell. Premenné je možné chrániť pred neúmyselným prepísaním ich nastavením ako určených len na čítanie nasledujúcim spôsobom:

```
readonly TMOUT TIMEOUT
```

Premenné prostredia TMOUT a TIMEOUT sa nastavujú v súboroch .profile užívateľov alebo v súbore /etc/security/.profile. Toto umožňuje pridanie súboru do súboru .profile užívateľa pri vytvorení užívateľa.

Ochrana Stack Execution Disable

Zachovane zabezpečenia počítačových systémov tvorí dôležité hľadisko podnikania On Demand. V dnešnom svete s husto zosieťovanými prostrediami sa stalo mimoriadnou výzvou odrážanie útokov z rôznych zdrojov.

Zvyšuje sa pravdepodobnosť, že počítačové systémy padnú za obeť dômyselným útokom, ktorých cieľom je zastavenie každodennej práce podnikov a vládnych agentúr. Pretože žiadne bezpečnostné opatrenie nedokáže zabezpečiť bezchybnú ochranu proti takýmto útokom, mali by ste do boja proti týmto útokom nasadiť viacero bezpečnostných mechanizmov. Táto časť sa zaoberá bezpečnostným mechanizmom, ktorý sa používa so systémom AIX, aby odrazil útoky vďaka spusteniu na báze pretečenia vyrovnávacej pamäte.

Narušenia bezpečnosti sa vyskytujú v mnohých formách, ale jednou z najbežnejších metód je monitorovanie systémom poskytnutých administračných nástrojov, vyhľadanie a zneužitie pretečení vyrovnávacích pamätí. Útoky pri pretečení vyrovnávacej pamäte sa vyskytujú pri prepísaní vyrovnávacej pamäte interného programu, pretože platnosť údajov nebola náležite overená (ako napríklad príkazový riadok, premenná prostredia, I/O diskov alebo terminálov). Kód útočníka sa vloží do spusteného procesu prostredníctvom pretečenia vyrovnávacej pamäte, čím sa zmení cesta spustenia spusteného procesu. Návratová adresa sa prepíše a presmeruje do umiestnenia vloženého kódu. K bežným príčinám prieniku do systému patrí nesprávna alebo neexistujúca kontrola hraníc alebo nesprávne predpoklady o platnosti zdrojov údajov. Napríklad, pretečenie vyrovnávacej pamäte môže nastať, keď je údajový objekt dostatočne veľký, aby pojal 1 KB údajov, ale program nekontroluje hranice vstupu, a preto sa môže stať, že do tohto údajového objektu sa skopíruje viac ako 1 KB.

Cieľom narušiteľa je zaútočiť na príkaz a/alebo nástroj, ktorý poskytuje privilégia typu root bežnému užívateľovi. Riadenie programu sa získava so všetkými zapnutými privilégiami povoľujúcimi pretečenie vyrovnávacích pamätí. Útoky sa zvyčajne sústreďujú na sadu UID vlastníka s oprávneniami typu root alebo na programy, ktoré vedú k spusteniu prostredia Shell, a takto aj k získaniu prístupu prostredia Shell typu root do systému.

Týmto útokom môžete zamedziť, keď zablokujete spustenie kódu útočníka, ktorý je zadávaný prostredníctvom pretečenia vyrovnávacej pamäte. Zakážte spustenie v pamäťových oblastiach procesu, v ktorých sa spúšťanie zvyčajne neodohráva (zásobníkové a haldové pamäťové oblasti).

Ochranný mechanizmus SED pre pretečenie vyrovnávacej pamäte:

Systém AIX povolil mechanizmus SED (Stack Execution Disable), aby zakázal spustenie kódu v oblastiach procesu pre zásobníky a výber údajov.

Zakázanie spustenia a následné ukončenie narušiteľského programu zamedzí útočníkovi získať privilégia užívateľa s oprávneniami typu root prostredníctvom útoku na pretečenú vyrovnávaciu pamäť. Hoci táto funkcia nezastaví pretečenia vyrovnávacej pamäte, poskytuje ochranu pred útokmi pomocou zákazu spúšťania v pretečených vyrovnávacích pamätiach.

S príchodom rodiny procesorov POWER4 môžete pre pamäť používať funkciu povolenia a/alebo zakázania spúšťania na úrovni stránok. Mechanizmus AIX SED používa túto východiskovú hardvérovú podporu pre implementáciu funkcie nespúšťania v oblastiach pamäte pre výber. Akonáhle je táto funkcia povolená, operačný systém počas spúšťania programov skontroluje a príznakom označí rôzne súbory. Potom pošle výstrahu správcovi pamäte operačného systému a manažérom procesov, že pre práve vytváraný proces je povolený SED. Oblasti pamäte pre výber budú označené pre nespúšťanie. Ak sa v týchto označených oblastiach vyskytne akékoľvek spustenie, hardvér vygeneruje príznak výnimky a operačný systém zastaví zodpovedajúci proces. Podrobnosti o výnimke a ukončení aplikácie sa zhromažďujú prostredníctvom udalostí chybového protokolu systému AIX.

SED sa implementuje hlavne prostredníctvom príkazu **sedmgr**. Príkaz **sedmgr** povoľuje riadenie celosystémového prevádzkového režimu SED ako aj nastavenie príznakov SED založených na súbore spustiteľného programu.

Režimy a monitorovanie SED:

Mechanizmus SED (Stack Execution Disable) je v systéme AIX implementovaný prostredníctvom príznakov celosystémového režimu a rovnako aj pomocou jednotlivých príznakov hlavičky spustiteľných súborov.

Zatiaľ čo celosystémové príznaky riadia celosystémovú prevádzku SED, príznaky úrovne súborov indikujú ako by sa malo v SED so súbormi zaobchádzať. Mechanizmus BOP (Buffer Overflow Protection) zabezpečuje štyri celosystémové prevádzkové režimy:

off Mechanizmus SED je vypnutý a žiadny proces nie je označený pre ochranu SED.

select Iba sada súborov select sa monitoruje a má povolenú ochranu SED. Súbory do sady súborov select sa vyberajú na základe revízie príznakov súvisiacich so SED v binárnych hlavičkách spustiteľného programu. Hlavička spustiteľného programu povoľuje, aby príznaky súvisiace so SED požadovali zaradenie do režimu **select**.

setidfiles
Povoľuje zapnúť SED nielen pre súbory požadujúce takýto mechanizmus, ale aj pre všetky dôležité systémové súbory **setuid** a **setgid**. V tomto režime operačný systém poskytuje SED pre súbory so sadou SED príznakov **request** a okrem toho aj povolí SED pre spustiteľné súbory s nasledujúcimi charakteristikami (okrem súborov, ktoré sú vo svojich súborových hlavičkách označené pre *exempt*):

- Súbory SETUID, ktoré sú vo vlastníctve typu root
- Súbory SETGID s primárnou skupinou **system** alebo **security**

all Ochranu SED majú všetky do systému zavedené spustiteľné programy okrem súborov, ktoré požadujú vyňatie z režimu SED. Príznaky súvisiace s vyňatím sú súčasťou hlavičiek spustiteľných programov.

Funkcia SED v systéme AIX poskytuje aj schopnosť monitorovania procesu namiesto jeho zastavenia, keď nastane výnimka. Toto celosystémové riadenie povoľuje administrátorovi systému skontrolovať poruchy a problémy v systémovom prostredí jeho monitorovaním pred umiestnením SED vo výrobných systémoch.

Príkaz **sedmgr** poskytuje voľbu, ktorá, keď nastane výnimka, umožňuje zapnúť SED na monitorovanie súborov namiesto zastavenia daných procesov. Správca systému môže zhodnotiť, či spustiteľný program vykonáva právoplatné spúšťanie zásobníkov. Toto nastavenie funguje v spojení s celosystémovým režimom, nastaveným pomocou voľby -c. Keď je režim **monitor** zapnutý, systém povolí procesu pokračovať v prevádzke, aj keď nastane výnimka týkajúca sa SED. Operačný systém namiesto zastavenia procesu zaprotokoluje výnimku do chybového protokolu AIX. Ak je SED monitorovanie vypnuté, operačný systém zastaví každý narušiteľský proces a vygeneruje výnimku na zariadení SED.

Aby sa mohli zmeny pre celosystémové príznaky režimu SED prejaviť, vyžaduje sa reštart systému. Všetky tieto typy udalostí sa auditujú.

Príznaky SED pre spustiteľné programy:

V systéme AIX môžete príkaz **sedmgr** použiť na označenie spustiteľných programov, z mechanizmu SE, príznakom.

Spojovací program bol vylepšený tak, aby podporoval dva nové príznaky, súvisiace so SED, pre povolenie volieb select a exempt v hlavičkách spustiteľného programu. Príznak **select** povoľuje spustiteľný program s cieľom požiadať a byť súčasťou ochrany SED počas režimu select celosystémovej operácie SED, kým príznak **exempt** povoľuje spustiteľný program s cieľom požiadať o vyňatie z mechanizmu SED. Tieto spustiteľné programy nemajú povolené zakázanie spustenia v žiadnej pamäťovej oblasti procesu.

Príznak vyňatia povoľuje administrátorovi systému monitorovať mechanizmus SED a vyhodnotiť situáciu. Podľa potreby, môže správca systému povoliť spustenie aplikácií v zásobníkových a údajových oblastiach, pričom si uvedomuje riziko s tým spojené.

V nasledujúcej tabuľke uvidíte, ako celosystémové nastavenia a súborové nastavenia vplývajú na prevádzkový režim SED:

Tabuľka 2. Celosystémové nastavenia a súborové nastavenia vplývajúce na režim SED

	Príznaky SED spustiteľných súborov			
Režim systému SED	**request**	**exempt**	**system**	**Súbory setuid-root alebo setgid-system/security**
off	–	–	–	–
select	enabled	–	–	–
setgidfiles	enabled	–	–	enabled
all	enabled	–	enabled	enabled

Problémy a hľadiská SED:

AIX SED sa štandardne dodáva v režime **select**. Množstvo programov **setuid** a **setgid** má pre SED povolený režim **select** a štandardne pracuje v chránenom režime.

Povolenie SED môže spôsobiť prerušenie starších binárnych súborov, ak nebudú schopné ošetriť funkciu nespúšťania v oblastiach zásobníkových háld. Tieto aplikácie musia byť spustené v oblastiach zásobníkových údajov. Administrátor systému môže vyhodnotiť situáciu a označiť príznakom súbor určený na vyňatie pomocou príkazu **bopmgr**. AIX Java™ 1.3.1 a AIX Java 1.4.2 majú kompilátory Just-In-Time (JIT), ktoré dynamicky generujú a spúšťajú natívny objektový kód počas spúšťania aplikácií Java (Java Virtual Machine rozhoduje, ktorý kód sa má kompilovať na základe profilu spustenia aplikácie). Tento objektový kód je uložený v údajových vyrovnávacích pamätiach, ktoré alokuje JIT. Následne, ak je AIX nakonfigurovaný, aby sa spustil v SED režime **ALL**, správca systému musí nastaviť príznak výnimky binárneho súboru Java.

Keď sa v spustiteľných súboroch zmenia príznaky súvisiace so SED, použijú sa len pre budúce zavedenie a spustenie súboru. Táto zmena sa nepoužije pre aktuálne prebiehajúce procesy na báze tohto súboru. Zariadenie SED riadi a monitoruje 32-bitové aj 64-bitové spustiteľné programy pre celosystémové nastavenia a nastavenie súborovej úrovne. Zariadenie SED je k dispozícii iba vtedy, keď sa operačný systém AIX používa so 64-bitovým jadrom.

Súvisiace informácie

príkaz **sedmgr**

Zariadenie na protokolovanie chýb systému AIX

Zvládnutie starostí s X11 a CDE

So serverom X11 X a Common Desktop Environment (CDE) sa spája určité ohrozenie bezpečnosti.

Odstránenie súboru /etc/rc.dt:

Súbor /etc/rc.dt odstráňte v systémoch, ktoré vyžadujú vysokú úroveň zabezpečenia.

Hoci je činnosť rozhrania CDE pre užívateľov pohodlná, sú s ňou spojené bezpečnostné problémy. preto toto prostredie nespúšťajte na serveroch, ktoré vyžadujú vysokú úroveň zabezpečenia. Najlepším riešením je vyhnúť sa inštalácii množín súborov prostredia CDE (dt). Ak ste do svojho systému nainštalovali tieto sady súborov, pouvažujte o ich odinštalovaní, hlavne však o odinštalovaní skriptu /etc/rc.dt, ktorý spúšťa CDE.

Ďalšie informácie o prostredí CDE nájdete v príručke *Operating system and device management*.

Zamedzenie neoprávneného monitorovania vzdialeného servera X:

Dôležitým problémom so zabezpečením spojeným so serverom X11 je neoprávnené tiché monitorovanie vzdialeného servera.

Príkazy **xwd** a **xwud** sa dajú použiť na monitorovanie činnosti servera X, pretože dokážu zachytiť stlačenie klávesu, čo môžu viesť k prezradeniu hesiel a iných dôverných údajov. Ak chcete vyriešiť tento problém, odstráňte vyššie uvedené spustiteľné súbory, pokiaľ nie sú v danej konfigurácii potrebné. Alternatívnym spôsobom je zmena prístupu k týmto súborom len pre užívateľov s oprávneniami typu root.

Príkazy **xwd** a **xwud** sa nachádzajú v sade súborov X11.apps.clients.

Ak príkazy **xwd** a **xwud** potrebujete, zvážte použitie aplikácie OpenSSH alebo MIT Magic Cookies. Tieto aplikácie tretích strán pomáhajú zamedziť rizikám, ktoré vznikajú pri spustení príkazov **xwd** a **xwud**.

Ďalšie informácie o OpenSSH a MIT Magic Cookies nájdete v príslušnej dokumentácii jednotlivých aplikácií.

Zapnutie a vypnutie riadenia prístupov:

Server X povoľuje hostiteľom používať príkaz **xhost +** na pripojenie k vášmu systému.

Názov hostiteľa je potrebné zadať spolu s príkazom **xhost +**, pretože tento príkaz zakazuje riadenie prístupu pre server X. Povoľuje to udeliť prístup k špecifickým hostiteľom, čo uľahčuje monitorovanie možných útokov na server X. Ak chcete udeliť prístup k určitému hostiteľovi, spustite príkaz **xhost** nasledovným spôsobom:

```
# xhost + hostname
```

Ak neuvediete názov hostiteľa, prístup bude udelený všetkým hostiteľom.

Bližšie informácie o príkaze **xhost** nájdete v *Commands Reference*

Vypnutie užívateľských povolení na spustenie príkazu xhost:

Neoprávnenému spusteniu príkazu **xhost** môžete zabrániť pomocou príkazu **chmod**.

Iným spôsobom na zaistenie správneho používania príkazu **xhost** je zakázať obmedziť spúšťanie tohto príkazu len pre oprávnenie užívateľa s oprávneniami typu root. Ak to chcete vykonať, pomocou príkazu **chmod** zmeňte povolenia /usr/bin/X11/xhost na 744 takto:

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

Zoznam programov setuid/setgid

Na systéme AIX sa nachádza viacero programov setuid/setgid. Tieto privilégiá na príkazy, ktoré nemusia byť k dispozícii bežným užívateľom, môžete odstrániť.

Nasledujúce programy sa nainštalujú v rámci štandardnej inštalácie systému AIX. V systéme AIX konfigurovanom pre CC je tento zoznam prečistený a obsahuje menej programov.

- /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

Užívatelia, skupiny a heslá

Môžete riadiť užívateľov a skupiny systému AIX.

Automatické vytvorenie domovského adresára pri prihlásení

Operačný systém AIX môže automaticky vytvoriť domovský adresár pri prihlásení užívateľa.

Táto funkcia je užitočná pre vzdialene definovaných užívateľov (napríklad užívateľov definovaných na LDAP serveri), u ktorých je možné, že nemajú domovský adresár v lokálnom systéme. Operačný systém AIX poskytuje dva mechanizmy na automatické vytvorenie domovského adresára pri prihlásení užívateľa: štandardný mechanizmus systému AIX a mechanizmus PAM. Tieto mechanizmy môžu byť povolené súčasne.

Mechanizmus AIX

Mechanizmus AIX pokrýva prihlásenie prostredníctvom týchto príkazov: **getty**, **login**, **rlogin**, **rsh**, **telnet** a **tsm**. Mechanizmus AIX podporuje autentifikáciu STD_AUTH a autentifikáciu PAM_AUTH s použitím modulu pam_aix. Povoľte mechanizmus AIX v súbore **/etc/security/login.cfg** nastavením atribútu mkhomeatlogin v odseku usw na hodnotu true (pozrite si v súbore /etc/security/login.cfg ďalšie informácie o tomto súbore). Na zapnutie alebo vypnutie funkcie automatického vytvorenia domovského adresára pri prihlásení sa používa príkaz **chsec**. Ak chcete napríklad zapnúť funkciu, spustite tento príkaz:

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

Ak je funkcia zapnutá, proces prihlásenia po úspešnej autentifikácii kontroluje domovský adresár užívateľa. Ak domovský adresár užívateľa neexistuje, bude vytvorený.

Poznámka: Atribút **mkhomeatlogin** je podporovaný len v AIX, verzia 6.1 s technologickou úrovňou 6100-02 alebo vyššia.

Mechanizmus PAM

AIX poskytuje aj modul pam_mkuserhome na vytváranie domovských adresárov pre mechanizmy PAM. Modul pam_mkuserhome môže byť uložený do zásobníka s ostatnými modulmi relácií pre služby prihlásenia. Ak chcete pre službu zapnúť modul PAM, musíte do tejto služby pridať položku. Ak chcete napríklad zapnúť vytvorenie domovského adresára prostredníctvom príkazu **telnet** pomocou PAM, do súboru **/etc/pam.cfg** pridajte nasledujúcu položku:

```
telnet session optional pam_mkuserhome
```

ID konta

Každé konto užívateľa má numerický ID, ktorý ho označuje jedinečným spôsobom. Operačný systém AIX udeľuje autorizácie na základe identifikátora konta.

Je dôležité pochopiť, že kontá s rovnakým ID sú virtuálne tým istým kontom. Príkazy AIX **mkuser** a **mkgroup** pri vytváraní užívateľov a skupín vždy skontrolujú cieľový register, aby konto, ktoré má byť vytvorené, nemalo kolíziu ID s existujúcimi kontami.

Systém možno tiež nakonfigurovať počas vytvárania konta na kontrolu všetkých užívateľských (skupinových) registrov pomocou systémového atribútu **dist_uniqid**. Atribút **dist_uniqid** stanzy usw v súbore /etc/security/login.cfg možno riadiť príkazom **chsec**. Ak chcete nakonfigurovať systém, aby zakaždým kontroloval kolíziu ID voči všetkým registrom, spustite:

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

Pre atribút **dist_uniqid** existujú tri platné hodnoty:

never Táto hodnota nekontroluje kolíziu ID voči necieľovým registrom (predvolená hodnota).

always Táto hodnota kontroluje kolíziu ID voči ostatným registrom. Ak sa medzi cieľovým a niektorým iným registrom zistí kolízia, príkaz **mkuser** (**mkgroup**) vyberie jedinečný ID nepoužívaný žiadnym iným registrom. Zlyhanie nastane len vtedy, ak je hodnota ID zadaná z príkazového riadka (napríklad mkuser id=234 foo a užívateľ použil v niektorom z registrov ID 234).

uniqbyname
Táto hodnota kontroluje kolíziu ID voči ostatným registrom. Kolízia medzi registrami je povolená len vtedy, ak má konto, ktoré má byť vytvorené, rovnaký názov ako existujúce konto pre typ príkazu mkuser id=123 foo. Ak ID nie je zadaný z príkazového riadka, nové konto možno nemá rovnakú hodnotu ID ako existujúce konto s rovnakým názvom v inom registri. Napríklad *acct1* s ID 234 je lokálnym kontom. Pri vytváraní konta LDAP *acct1*, mkuser -R LDAP acct1 môže vybrať pre konto LDAP jedinečný ID 235. Výsledkom je *acct1* s ID 234 na lokálnom registri a *acct1* s 235 na LDAP.

Poznámka: Zisťovanie kolízie ID v cieľovom registri sa vykoná vždy bez ohľadu na atribút **dist_uniqid**.

Hodnota **uniqbyname** funguje dobre v prípade dvoch registrov. Pri viacerých registroch a v prípade, že medzi dvomi registrami už nastala kolízia ID, nie je pri vytváraní nového konta v treťom registri pomocou kolíznych hodnôt ID správanie **mkuser** (**mkgroup**) presne stanovené. Vytvorenie nového konta môže byť úspešné alebo neúspešné v závislosti od poradia kontroly registrov.

Predpokladajme, že systém je nakonfigurovaný napríklad s tromi registrami: lokálnym, LDAP a DCE. Konto *acct1* existuje v LDAP a konto *acct2* existuje v DCE, obe s ID 234. Keď administrátor systému spustí príkaz mkuser -R files id=234 acct1 (mkgroup -R files id=234 acct1) na vytvorenie lokálneho konta s hodnotou **uniqbyname**, príkaz **mkuser** (**mkgroup**) skontrolujte najprv register LDAP a zistí, že ID 234 je použité kontom LDAP *acct1*. Keďže konto, ktoré sa má vytvoriť, má rovnaký názov, príkaz **mkuser** (**mkgroup**) úspešne vytvorí lokálne konto *acct1* s ID 234. Ak sa skontroluje najprv register DCE, príkaz **mkuser** (**mkgroup**) zistí, že konto DCE *acct2* použilo ID 234 a vytvorenie lokálneho konta *acct1* nebude úspešné. Kontrola kolízie ID si vynúti jedinečnosť ID medzi lokálnym a vzdialenými registrami alebo medzi vzdialenými registrami. Medzi novo vytvoreným kontom na vzdialenom registri a existujúcimi lokálnymi užívateľmi na iných systémoch používajúcich rovnaký vzdialený register neexistuje garancia jedinečnosti ID. Ak vzdialený register nie je v čase spustenia príkazu **mkuser** (**mkgroup**) dosiahnuteľný, príkaz ho obíde.

Konto užívateľa s oprávneniami typu root

Konto užívateľa s oprávneniami typu root má prakticky neobmedzený prístup do všetkých programov, súborov a prostriedkov v systéme.

Konto užívateľa s oprávneniami typu root je špeciálny užívateľ zo súboru /etc/passwd s užívateľským ID (UID) 0 a všeobecne sa nazýva *root*. To, čo ho robí takým špeciálnym, nie je meno užívateľa, ale UID s hodnotou 0. To znamená, že ktorýkoľvek užívateľ, ktorý má UID s hodnotou 0 má tiež rovnaké privilégiá ako užívateľ s oprávneniami typu root. Rovnako, konto s oprávneniami typu root je vždy autentifikované pomocou lokálnych bezpečnostných súborov.

Konto typu root by vždy malo mať heslo, ktoré by sa nemalo zdieľať. Kontu typu root by sa malo prideliť heslo okamžite po nainštalovaní systému. Heslo konta typu root by mal poznať len správca systému. Na vykonávanie funkcií správy systému, ktoré vyžadujú privilégiá typu root, by správcovia systému mali pôsobiť ako užívateľ s oprávneniami typu root. Pri všetkých ostatných operáciách by sa mali vrátiť do svojho normálneho užívateľského konta.

Upozornenie: Bežné pôsobenie ako užívateľ s oprávneniami typu root by mohlo mať za následok poškodenie systému, lebo konto typu root obchádza mnohé ochrany v systéme.

Vypnutie priameho prihlásenia užívateľa root:

Bežnou metódou útoku počítačových pirátov je získanie hesla root.

Aby ste predišli tomuto typu útoku, môžete zakázať priamy prístup k vášmu ID užívateľa s oprávneniami typu root a potom požadovať, aby vaši správcovia systému získali privilégiá užívateľa s oprávneniami typu root pomocou príkazu **su -**. Okrem oprávnenia na odstránenie užívateľa s oprávneniami typu root ako bodu útoku, obmedzenie priameho prístupu užívateľa s oprávneniami typu root povoľuje monitorovať, ktorí užívatelia získali prístup s oprávneniami typu root a tiež čas ich akcie. Tieto informácie získate po zobrazení súboru /var/adm/sulog. Ďalšou možnosťou je povolenie auditovania systému, ktoré upozorní na tento typ činnosti.

Ak chcete zakázať prístup pomocou vzdialeného prihlásenia užívateľa s oprávneniami typu root, upravte súbor /etc/security/user. Na položke pre root zadajte False ako hodnotu rlogin.

Pred zakázaním vzdialeného prihlásenia užívateľa s oprávneniami typu root analyzujte a naplánujte riešenia pre situácie, ktoré by mohli znemožniť systémovému administrátorovi prihlásiť sa s ID užívateľa iného typu než root. Napríklad, ak domovský súborový systém užívateľa je plný, užívateľ by nemal byť schopný prihlásiť sa. Ak bolo vzdialené prihlasovanie root vypnuté a užívateľ, ktorý môže používať príkaz **su -** na zmenu na root, mal plný domáci systém súborov, root nebude môcť nikdy prevziať kontrolu nad systémom. Tomuto problému sa môžu systémoví administrátori vyhnúť vytvorením svojich domovských systémov súborov, ktoré budú väčšie než priemerné systémy súborov užívateľov.

Užívateľské kontá

Pre užívateľské kontá existuje niekoľko úloh administrácie bezpečnosti.

Odporúčané užívateľské atribúty:

Správa užívateľov pozostáva z vytvárania užívateľov a skupín a z definovania ich atribútov.

Najhlavnejším atribútom užívateľov je spôsob ich autentifikácie. Užívatelia predstavujú primárnych agentov v systéme. Ich atribúty riadia ich prístupové práva, prostredie, spôsob ich autentifikácie ako aj spôsob, čas a miesto prístupu na ich kontá.

Skupiny predstavujú súhrn užívateľov, ktorí zdieľajú rovnaké prístupové práva k chráneným prostriedkom. Skupina má svoje ID a skladá sa z členov a z administrátorov. Tvorca skupiny je zvyčajne prvý administrátor.

Pre každé konto užívateľa možno nastaviť mnohé atribúty, vrátane hesla a atribútov prihlásenia. Zoznam konfigurovateľných atribútov nájdete v časti “Prehľad systému kvót diskového priestoru” na strane 73. Odporúčajú sa nasledovné atribúty:

- Každý užívateľ by mal mať ID, ktoré nezdieľa so žiadnym iným užívateľom. Všetky bezpečnostné ochrany a zabezpečovacie nástroje budú fungovať, len ak každý užívateľ bude mať jedinečné ID.
- Užívateľom v systéme priraďujte užívateľské mená s určitým významom. Ideálne sú skutočné mená, lebo väčšina systémov elektronickej pošty používa ID užívateľa na označovanie doručovanej pošty.
- Užívateľov pridávajte, meňte a odstraňujte pomocou rozhrania nástroja SMIT. Aj keď môžete tieto úlohy vykonať aj z príkazového riadka, rozhranie nástroja SMIT vám pomôže vyhnúť sa drobným chybám.
- Užívateľskému kontu nepridelíujte úvodné heslo, kým sa užívateľ skutočne nechytá prihlásiť do systému. Ak je pole hesla definované ako * (hviezdička) v súbore /etc/passwd, informácie o konte budú ponechané, ale nikto sa na dané konto nebude môcť prihlásiť.
- Nemeňte systémom definované ID užívateľov, ktoré sú potrebné pre správnu funkčnosť systému. Zoznam systémom definovaných ID užívateľov sa nachádza v súbore /etc/passwd.
- Vo všeobecnosti, parameter *admin* nenastavujte pre žiadne ID užívateľov na hodnotu true. Iba užívateľ s oprávneniami typu root môže meniť atribúty pre užívateľov, ktorý majú admin=true nastavené v súbore /etc/security/user.

Operačný systém podporuje štandardné užívateľské atribúty, ktoré zvyčajne nájdete v súboroch /etc/passwd a /etc/system/group, napríklad:

Autentifikačné informácie
Určuje heslo

Oprávnenia
Určuje identifikátor užívateľa, hlavnú skupinu a ID doplnkovej skupiny

Prostredie
Určuje domovské alebo užívateľské prostredie

Obmedzenie dĺžky mena užívateľa a názvu skupiny:

Môžete nakonfigurovať a získať obmedzenie dĺžky mena užívateľa a názvu skupiny.

Predvolená hodnota parametra obmedzenia dĺžky mena užívateľa a názvu skupiny je 9 znakov. Pre systémy AIX 5.3 a novšie môžete zvýšiť limit dĺžky mena užívateľa a názvu skupiny z 9 na 256 znakov. Keďže parameter obmedzenia dĺžky mena užívateľa a názvu skupiny zahŕňa ukončovací znak NULL, skutočné platné dĺžky mien a názvov sú 8 až 255 znakov.

Obmedzenie dĺžky mena užívateľa a názvu skupiny je zadané pomocou parametra systémovej konfigurácie **v_max_logname** pre zariadenie sys0. Hodnotu parametra **v_max_logname** môžete zmeniť alebo získať z jadra alebo databázy ODM. Hodnota parametra v jadre je hodnota, ktorú systém používa počas spúšťania. Hodnota parametra v databáze ODM je hodnota, ktorú systém používa po nasledujúcom reštarte.

Poznámka: Ak po zvýšení znížite limit dĺžky mena užívateľa alebo názvu skupiny, môže nastať neočakávané správanie. Mená užívateľov a názvy skupín, ktoré ste vytvorili s vyšším limitom, sa môžu v systéme ešte stále nachádzať.

Získanie limitu dĺžky mena užívateľa a názvu skupiny z databázy ODM:

Pomocou príkazov alebo podprogramov môžete získať parameter **v_max_logname**.

Pomocou príkazu **lsattr** získate parameter **v_max_logname** v databáze ODM. Príkaz **lsattr** zobrazí parameter **v_max_logname** ako atribút max_logname.

Bližšie informácie nájdete v príkaze **lsattr** v časti *Commands Reference, Volume 3*.

Nasledujúci príklad ukazuje, ako pomocou príkazu **lsattr** získate atribút max_logname:

```
$ lsattr -El sys0
SW_dist_intr    false           Povoliť SW distribúciu prerušení                True
autorestart     true            Automaticky REBOOTOVAŤ systém po zlyhaní        True
boottype        disk            nie je k dispozícii                             False
capacity_inc    1.00            Prírastok kapacity procesora                    False
capped          true            Oddiel je uzavretý                              False
conslogin       enable          Prihlásenie na systémovú konzolu                False
cpuguard        enable          Sprievodca CPU                                  True
dedicated       true            Oddiel je vyhradený                             False
ent_capacity    4.00            Oprávnená kapacita procesora                    False
frequency       93750000        Frekvencia systémovej zbernice                  False
fullcore        false           Povoliť úplný výpis CORE                        True
fwversion       IBM,SPH01316    Verzia firmvéru a úrovne revízie                False
iostat          false           Nepretržite udržiavať históriu I/O DISKU        True
keylock         normal          Stav zámku kľúča v čase zavedenia               False
max_capacity    4.00            Maximálna potenciálna kapacita procesora        False
max_logname     20              Maximálna dĺžka prihlásenia v čase zavedenia    True
maxbuf          20              Max. počet stránok v blok. CACHE I/O VYR. PAMÄTE True
maxmbuf         0               Maximum kbajtov skutočnej pam. povolený pre MBUFS True
maxpout         0               Zn. VYSOKEJ hlad. pre čakajúci zápis I/O na 1 súbor  True
maxuproc        128             Maximálny počet PROCESOV povolených na užívateľa  True
```

```
min_capacity    1.00            Minimálna potenciálna kapacita procesora          False
minpout         0               Zn. NÍZKEJ hlad. pre čakajúci zápis I/O na 1 súbor  True
modelname       IBM,7044-270    Názov počítača                                    False
ncargs          6               Veľkosť zoznamu ARG/ENV v 4-kbajtových blockoch   True
pre430core      false           Použiť výpis JADRA štýlu pre-430                  True
pre520tune      disable         Režim kompatibility ladenia pre-520               True
realmem         3145728         Množstvo použiteľnej fyzickej pamäte v kB         False
rtasversion     1               Verzia RTAS otvoreného firmvéru                   False
sec_flags       0               Bezpečnostné príznaky                             True
sed_config      select          Režim SED (Stack Execution Disable)               True
systemid        IBM,0110B5F5F   Systémový identifikátor hardvéru                  False
variable_weight 0               Závažnosť kapacity premenného procesora           False
$
```

Získanie limitu dĺžky mena užívateľa a názvu skupiny z jadra:

Pomocou príkazov a podprogramov môžete získať parameter **v_max_logname** z jadra.

Pomocou príkazu getconf

Pomocou príkazu **getconf** s parametrom **LOGIN_NAME_MAX** získate limit dĺžky mena užívateľa a názvu skupiny v jadre. Výstup príkazu **getconf** obsahuje ukončovací znak NULL.

Nasledujúci príklad ukazuje, ako príkaz **getconf** získa aktuálny limit dĺžky mena užívateľa a názvu skupiny z jadra:

```
$ getconf LOGIN_NAME_MAX
20
$
```

Pomocou podprogramu sysconf

Pomocou podprogramu sysconf s parametrom **_SC_LOGIN_NAME_MAX** získate limit dĺžky mena užívateľa a názvu skupiny v jadre.

Nasledujúci príklad ukazuje, ako pomocou podprogramu sysconf získate aktuálny limit dĺžky mena užívateľa a názvu skupiny z jadra:

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

       len = sysconf(_SC_LOGIN_NAME_MAX);

       printf("Limit dĺžky mena/názvu je %d\n", len);
}
```

Pomocou podprogramu sys_parm

Pomocou podprogramu sys_parm s parametrom **SYSP_V_MAX_LOGNAME** získate aktuálny limit dĺžky mena užívateľa v jadre.

Nasledujúci príklad ukazuje, ako pomocou podprogramu sys_parm získate aktuálny limit dĺžky mena užívateľa z jadra:

```
#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_dlzka_prihl_mena = %d\n", myvar.v.v_max_logname.value);
   else
      printf("sys_parm() zlyhal rc = %d, číslo chyby = %d\n", rc, errno);
}
```

Zmena limitu pre dĺžku názvu užívateľa a skupiny v databáze ODM:

Hodnotu limitu pre dĺžku názvu užívateľa a skupiny v kerneli môžete zmeniť iba počas fázy zavedenia systému. Hodnotu v databáze ODM zmeníte príkazom **chdev**. Zmeniť bude účinná po najbližšom reštartovaní systému.

Nasledujúci príklad ukazuje, ako príkazom **chdev** zmeníte parameter **v_max_logname** v databáze ODM:

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

Riadenie užívateľských kont:

Užívateľské kontá majú atribúty, ktoré sa dajú zmeniť.

Ku každému kontu užívateľa je priradená skupina atribútov. Tieto atribúty sa tvoria zo štandardných hodnôt, keď je užívateľ vytvorený pomocou príkazu **mkuser**. Atribúty možno zmeniť príkazom **chuser**. Toto sú používateľské atribúty, ktoré riadia prihlasovanie a nemajú vzťah ku kvalite hesiel:

account_locked
Ak musí byť konto explicitne uzamknuté, môžete tento atribút nastaviť na hodnotu True; štandardná hodnota je False.

admin Ak je nastavený na hodnotu True, tento užívateľ nemôže zmeniť heslo. Zmeniť ho môže len administrátor.

admgroups
Obsahuje zoznam skupín, pre ktoré má užívateľ administratívne práva. V týchto skupinách môže užívateľ pridávať alebo mazať členov.

auth1 Metóda autentifikácie používaná na udelenie prístupu užívateľovi. Obvykle je nastavený na hodnotu SYSTEM a spôsobí použitie novších metód.

Poznámka: Atribút **auth1** je odmietnutý a už by nemal byť použitý.

auth2 Metóda, ktorá sa spúšťa po autentifikácii užívateľa spôsobom zadaným v atribúte **auth1**. Tento atribút neumožňuje blokovanie prístupu k systému. Obvykle je nastavený na hodnotu NONE.

Poznámka: Atribút **auth2** je odmietnutý a už by nemal byť použitý.

daemon
Tento boolovský parameter určuje, či užívateľ môže spustiť démonov alebo subsystémy pomocou príkazu **startsrc**. Takisto obmedzuje použitie zariadení cron a at.

login Špecifikuje, či sa tento užívateľ môže prihlasovať. Úspešné prihlásenie resetuje atribút **unsuccessful_login_count** na hodnotu 0 (zo subrutiny **loginsuccess**).

logintimes
Obmedzuje čas prihlásenia užívateľa. Napríklad, užívateľ by mohol mať obmedzený prístup k systému len počas normálnych pracovných hodín.

registry
Určuje register užívateľa. môže sa použiť, aby povedal systému o náhradných registroch pre užívateľské informácie, ako napríklad NIS, LDAP alebo Kerberos.

rlogin Určuje, či sa určený užívateľ môže prihlásiť pomocou príkazu **rlogin** alebo **telnet**. Atribút rlogin riadi iba vzdialené prihlásenie. Informácie o riadení schopnosti spúšťať jednotlivé vzdialené príkazy nájdete v téme rcmds.

su Určuje, či sa iní užívatelia môžu prepnúť na toto ID príkazom **su**.

sugroups
Určuje, ktorým skupinám je povolené prepínať sa na toto ID užívateľa.

ttys Ohraničuje isté kontá na fyzicky bezpečné oblasti.

expires Spravuje študentské alebo hosťovské kontá. Možno ho použiť aj na dočasné vypnutie kont.

loginretries
Určuje maximálny počet následných neúspešných pokusov o prihlásenie pred tým, ako systém zamkne ID užívateľa. Neúspešné pokusy sa zaznamenávajú v súbore /etc/security/lastlog.

umask Určuje úvodný atribút **umask** pre užívateľa.

rcmds Určuje, či určený užívateľ môže spúšťať jednotlivé príkazy prostredníctvom príkazu **rsh** alebo **rexec**. Hodnota allow určuje, že užívateľ môže spúšťať tieto príkazy prostredníctvom príkazov **rsh** a **rexec**. Hodnota deny určuje, že užívateľ nemôže vzdialene spúšťať príkazy. Hodnota hostlogincontrol určuje, že spúšťanie vzdialených príkazov sa riadi podľa atribútov **hostallowedlogin** a **hostsdeniedlogin**. Informácie o riadení vzdialeného prihlasovania nájdete v téme o atribúte rlogin.

hostallowedlogin
Špecifikuje hostiteľov, ktorí danému používateľovi povoľujú prihlásiť sa. Tento atribút je určený na použitie v sieťovom prostredí, kde atribúty používateľov zdieľa viacero hostiteľov.

hostsdeniedlogin
Špecifikuje hostiteľov, ktorí danému používateľovi nepovoľujú prihlásiť sa. Tento atribút je určený na použitie v sieťovom prostredí, kde atribúty používateľov zdieľa viacero hostiteľov.

maxulogs
Určuje maximálny počet prihlásení na jedného používateľa. Ak užívateľ dosiahol maximálny počet povolených prihlásení, prihlásenie bude odmietnuté.

Celá sada užívateľských atribútov je definovaná v súboroch /etc/security/user, /etc/security/limits, /etc/security/audit/config a /etc/security/lastlog. Štandardná hodnota pre vytvorenie užívateľa pomocou príkazu **mkuser** je špecifikovaná v súbore /usr/lib/security/mkuser.default. V súbore mkuser.default musia byť špecifikované iba voľby, ktoré vo všeobecných statiach súborov /etc/security/user a /etc/security/limits vyraďujú všeobecné štandardné hodnoty a rovnako tam musia byť špecifikované aj triedy auditu. Niekoľko týchto atribútov riadi, ako sa užívateľ môže prihlásiť a možno ich nakonfigurovať tak, aby za určených okolností automaticky zamkli užívateľské konto (zabránili ďalšiemu prihlasovaniu).

Keď systém uzamkne užívateľské konto kvôli počtu neúspešných pokusov o prihlásenie sa, užívateľ sa nebude môcť prihlásiť, kým správca systému neresetuje užívateľský atribút **unsuccessful_login_count** v súbore /etc/security/lastlog na hodnotu, ktorá bude menšia ako hodnota opakovania prihlásenia. To možno vykonať nasledovne pomocou príkazu **chsec**:

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

Predvolené hodnoty možno zmeniť pomocou príkazu **chsec** s cieľom upraviť predvolenú stanzu v príslušnom súbore bezpečnosti, napríklad súbore /etc/security/user alebo /etc/security/limits. Mnohé predvolené hodnoty sú definované tak, aby predstavovali štandardné správanie. Ak si želáte atribúty, ktoré sa nastavujú pri vytváraní každého nového používateľa, zadať explicitne, zmeňte položku *user* v súbore /usr/lib/security/mkuser.default.

Informácie o rozšírených atribútoch hesiel užívateľov nájdete v časti “Heslá” na strane 62.

Prihlasovacie príkazy ovplyvnené užívateľskými atribútmi

Nasledujúca tabuľka vypisuje atribúty, ktoré riadia prihlasovanie a príslušné príkazy.

Užívateľský atribút	Príkazy
account_locked	**rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet**, **ftp**, **login**
login	Ovplyvňuje len prihlasovanie z konzoly. Hodnota atribútu **login** nemá vplyv na príkazy vzdialeného prihlasovania, príkazy vzdialeného prostredia shell alebo príkazy na vzdialené kopírovanie **rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet** a **ftp**.
logintimes	**rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet**, **ftp**, **login**
rlogin	Má vplyv len na príkazy vzdialeného prihlasovania, niektoré príkazy vzdialeného prostredia shell a určité príkazy vzdialeného kopírovania (**ssh**, **scp rlogin**, a **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 and 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**

Poznámka: **rsh** len zakazuje spustenie vzdialených príkazov. Vzdialené prihlásenia sú ešte stále povolené.

Súvisiace informácie:

Podrutina loginsuccess

Príkaz **rexec**

Príkaz **rsh**

Príkaz **startsrc**

Príkaz **su**

Prihlasovacie ID užívateľov:

Operačný systém identifikuje užívateľov podľa ich prihlasovacieho ID užívateľa.

Prihlasovacie ID užívateľa umožňuje systému sledovať všetky akcie užívateľa až ku ich zdroju. Po prihlásení užívateľa do systému, no pred spustením počiatočného programu užívateľa systém nastaví prihlasovacie ID procesu podľa ID užívateľa, ktoré našiel v databáze daného užívateľa. Všetky nasledovné procesu počas relácie prihlásenia sú označené týmto ID. Tieto značky predstavujú stopu všetkých činností vykonaných pod daným prihlasovacím ID užívateľa. Užívateľ môže počas relácie resetovať ID efektívneho užívateľa, ID reálneho užívateľa, ID efektívnej skupiny, ID reálnej skupiny a ID doplnkovej skupiny, ale nemôže zmeniť prihlasovacie ID užívateľa.

Posilnenie bezpečnosti užívateľov pomocou zoznamov prístupových práv:

Ak chcete dosiahnuť primeranú úroveň zabezpečenia v systéme, vytvorte konzistentnú politiku zabezpečenia pre riadenie kont užívateľov. Najbežnejšie používaný bezpečnostný mechanizmus je ACL (Access Control List).

Informácie o zoznamoch prístupových práv a bezpečnostnej politike nájdete v časti “Zoznamy riadenia prístupov (ACL)” na strane 115.

Premenná prostredia PATH:

Premenná prostredia **PATH** je dôležitým bezpečnostným prvkom. Špecifikuje ktoré adresáre sa majú prehľadať za účelom nájdenia príkazu.

Štandardná celosystémová hodnota **PATH** je špecifikovaná v súbore /etc/profile a každý užívateľ má zvyčajne hodnotu **PATH** v užívateľskom súbore $HOME/.profile. Hodnota **PATH** v súbore .profile buď vyradí celosystémovú hodnotu **PATH** alebo do nej pridá dodatočné adresáre.

Neoprávnené zmeny pre premennú prostredia **PATH** môžu užívateľovi v systéme povoliť "balamutenie" ostatných užívateľov (vrátane užívateľov s oprávneniami typu root). *Balamutiace* programy (nazývajú sa aj programy *Trójske kone*) zamieňajú systémové príkazy, a potom zachytávajú informácie určené pre takýto príkaz, ako napríklad užívateľské heslá.

Napríklad, predpokladajme, že užívateľ zmení hodnotu **PATH** tak, aby pri spustení príkazu systém najprv prehľadal adresár /tmp. Potom užívateľ vloží do adresára /tmp program s názvom **su**, ktorý bude žiadať heslo typu root presne ako príkaz **su**. Potom program **/tmp/su** pošle heslo typu root poštou užívateľovi a pred ukončením zavolá skutočný príkaz **su**. V tomto prípade by každý užívateľ s oprávneniami typu root, ktorý používa príkaz **su** odhalil svoje heslo, pričom by si toho vôbec nebol vedomý.

Na zabránenie akýmkoľvek problémom s premennou prostredia **PATH** pre správcov systému a užívateľov, vykonajte nasledovné kroky:

- Ak si nie ste istí, zadávajte úplné názvy ciest. Ak je zadaný úplný názov cesty, premenná prostredia **PATH** je ignorovaná.
- Do hodnoty premennej **PATH** zadanej pre užívateľa s oprávneniami typu root nikdy nezadávajte aktuálny adresár (určený **.** (bodkou)). Zabezpečte, aby v súbore /etc/profile nebol nikdy zadaný aktuálny adresár.
- Užívateľ s oprávneniami typu root by mal mať svoju vlastnú špecifikáciu **PATH** vo svojom súkromnom súbore .profile. Obvykle špecifikácia v súbore /etc/profile uvádza minimálny štandard pre všetkých užívateľov, zatiaľ čo užívateľ s oprávneniami typu root by mohol potrebovať viac alebo menej adresárov, ako je predvolené.
- Upozornite ostatných užívateľov, aby bez konzultácie so systémovým administrátorom nemenili obsah svojich súborov .profile. V opačnom prípade môže nepozorný užívateľ vykonať zmeny, ktoré umožnia nežiaduci prístup. Súbor užívateľa .profile by mal mať oprávnenia nastavené na 740.
- Správcovia systému by nemali používať príkaz **su** na získanie privilégií typu root z užívateľskej relácie, pretože je v platnosti užívateľská hodnota **PATH**, špecifikovaná v súbore .profile. Užívatelia si môžu nastaviť svoje vlastné súbory .profile. Správcovia systému by sa mali na užívateľský počítač prihlasovať ako užívateľ s oprávneniami typu root alebo by mali radšej použiť svoje vlastné ID a potom použiť nasledujúci príkaz:

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

  Týmto krokom sa zabezpečí, že počas relácie sa použije prostredie typu root. Ak správca systému nepracuje v inej užívateľskej relácii ako užívateľ s oprávneniami typu root, mal by počas relácie zadávať názvy úplných ciest.
- Premennú prostredia **IFS** (Input Field Separator) chráňte tak, aby nebola zmenená v súbore /etc/profile. Premenná prostredia **IFS** v súbore .profile sa dá použiť na zmenu hodnoty **PATH**.

Používanie démona secldapclntd:

Démon **secldapclntd** dynamicky riadi pripojenia na server LDAP.

Pri spustení sa démon **secldapclntd** pripojí na servery definované v súbore **/etc/security/ldap/ldap.cfg** (jedno pripojenie na server LDAP). Neskôr, ak démon **secldapclntd** určí, že pripojenie LDAP obmedzuje požiadavky spracovania LDAP, automaticky vytvorí ďalšie pripojenie na aktuálny server LDAP. Tento proces bude pokračovať, kým sa nedosiahne preddefinovaný maximálny počet pripojení. Po dosiahnutí maximálneho počtu pripojení nebudú pridané žiadne nové pripojenia.

Démon **secldapclntd** pravidelne kontroluje všetky pripojenia na aktuálny server LDAP. Ak je niektoré pripojenie s výnimkou prvého pripojenia nečinné počas preddefinovaného časového úseku, démon ho ukončí.

Premenná connectionsperserver v súbore **/etc/security/ldap/ldap.cfg** sa používa ako maximálny počet pripojení. Ak je však premenná connectionsperserver väčšia ako premenná numberofthread, démon **secldapclntd** nastaví hodnotu connectionsperserver na hodnotu numberofthread. Platné hodnoty pre premennú connectionsperserver sú 1 až 100. Predvolená hodnota je 10 (connectionsperserver: 10).

Premenná connectionmissratio v súbore **/etc/security/ldap/ldap.cfg** nastaví kritériá na vytváranie nových pripojení LDAP. Premenná connectionmissratio predstavuje percentuálny podiel operácií, ktoré pri získavaní pripojení LDAP (handle-miss) zlyhali počas prvých pokusov. Ak je počet neúspešných pokusov väčší ako premenná connectionmissratio, démon **secldapclntd** zlepší dotazy LDAP vytvorením nových pripojení LDAP (aby nepresiahli počet pripojení definovaný v premennej connectionsperserver). Platné hodnoty pre premennú connectionmissratio sú 10 až 90. Predvolená hodnota je 50 (connectionmissratio: 50).

Premenná connectiontimeout v súbore **/etc/security/ldap/ldap.cfg** sa používa ako časový úsek, počas ktorého môžu pripojenia ostať nečinné predtým, než ich démon **secldapclntd** zatvorí. Platné hodnoty pre premennú connectiontimeout sú 5 sekúnd a viac (žiadny maximálny limit). Predvolená hodnota je 300 sekúnd (connectiontimeout: 300).

Anonymné pripojenie k FTP s nastavením konta bezpečného užívateľa

Môžete nastaviť anonymné pripojenie k FTP so zabezpečeným užívateľským kontom.

Tento scenár nastavuje anonymné pripojenie k FTP s kontom bezpečného užívateľa pomocou rozhrania príkazového riadka a skriptu.

1. Zadaním nasledujúceho príkazu skontrolujte, či je v systéme nainštalovaná množina súborov bos.net.tcp.client:
   ```
   lslpp -L | grep bos.net.tcp.client
   ```
   Ak sa nezobrazí žiadny výstup, množina súborov nie je nainštalovaná. Inštrukcie o jej inštalácii nájdete v časti *Installation and migration*.
2. S oprávnením užívateľa typu root prejdite do adresára /usr/samples/tcpip. Napríklad:
   ```
   cd /usr/samples/tcpip
   ```
3. Spustením nasledujúceho skriptu nastavte konto:
   ```
   ./anon.ftp
   ```
4. Po zobrazení výzvy Are you sure you want to modify /home/ftp? zadajte yes. Zobrazí sa výstup podobný nasledujúcemu:
   ```
   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. Prejdite do adresára /home/ftp. Napríklad:
   ```
   cd /home/ftp
   ```
6. Zadaním nasledujúceho príkazu vytvorte podadresár home:
   ```
   mkdir home
   ```
7. Zadaním nasledujúceho príkazu zmeňte oprávnenia adresára /home/ftp/home na drwxr-xr-x:
   ```
   chmod 755 home
   ```
8. Zadaním nasledujúceho príkazu prejdite do adresára/home/ftp/etc:
   ```
   cd /home/ftp/etc
   ```
9. Zadaním nasledujúceho príkazu vytvorte podadresár objrepos:
   ```
   mkdir objrepos
   ```
10. Zadaním nasledujúceho príkazu zmeňte oprávnenia adresára /home/ftp/etc/objrepos na drwxrwxr-x:
    ```
    chmod 775 objrepos
    ```
11. Zadaním nasledujúceho príkazu zmeňte vlastníka a skupinu adresára /home/ftp/etc/objrepos na užívateľa typu root a systémovú skupinu:
    ```
    chown root:system objrepos
    ```
12. Vytvorte podadresár security napísaním
    ```
    mkdir security
    ```

13. Zmeňte oprávnenia adresára /home/ftp/etc/security na drwxr-x---. Použite tento príkaz:
    ```
    chmod 750 security
    ```
14. Pomocou nasledujúceho príkazu zmeňte vlastníka a skupinu adresára /home/ftp/etc/security na užívateľa typu root a systémovú skupinu:
    ```
    chown root:security security
    ```
15. Zadaním nasledujúceho príkazu prejdite do adresára /home/ftp/etc/security:
    ```
    cd security
    ```
16. Zadaním tejto rýchlej cesty nástroja SMIT pridajte užívateľa:
    ```
    smit mkuser
    ```

    Pre potreby tohto scenára sa pridáva užívateľ s názvom test.
17. Do polí SMIT zadajte tieto hodnoty:
    ```
    User NAME                                          [test]
    ADMINISTRATIVE USER?                                true
    Primary GROUP                                      [staff]
    Group SET                                          [staff]
    Another user can SU TO USER?                        true
    HOME directory                                     [/home/test]
    ```

    Po zadaní zmien stlačte kláves Enter, čím sa vytvorí nový užívateľ. Nástroj SMIT ukončite po dokončení jeho procesu.
18. Pomocou nasledujúceho príkazu vytvorte heslo pre tohto užívateľa:
    ```
    passwd test
    ```

    Po zobrazení výzvy zadajte požadované heslo. Nové heslo je potrebné zadať ešte raz pre potvrdenie.
19. Zmeňte na adresár /home/ftp/etc napísaním
    ```
    cd /home/ftp/etc
    ```
20. Pomocou nasledujúceho príkazu skopírujte obsah súboru /etc/passwd do súboru /home/ftp/etc/passwd:
    ```
    cp /etc/passwd /home/ftp/etc/passwd
    ```
21. Súbor /home/ftp/etc/passwd upravte pomocou obľúbeného editora. Napríklad:
    ```
    vi passwd
    ```
22. Zo skopírovaného obsahu odstráňte všetky riadky okrem riadkov pre užívateľa typu root, ftp a testovacích užívateľov. Po úprave by sa mal obsah podobať tomuto:
    ```
    root:!:0:0::/:/bin/ksh
    ftp:*:226:1::/home/ftp:/usr/bin/ksh
    test:!:228:1::/home/test:/usr/bin/ksh
    ```
23. Uložte zmeny a ukončite editor.
24. Zadaním nasledujúceho príkazu zmeňte oprávnenia súboru /home/ftp/etc/passwd na -rw-r--r--:
    ```
    chmod 644 passwd
    ```
25. Pomocou nasledujúceho príkazu zmeňte vlastníka a skupinu súboru /home/ftp/etc/passwd na užívateľa typu root a skupinu zabezpečenia:
    ```
    chown root:security passwd
    ```
26. Pomocou nasledujúceho príkazu skopírujte obsah súboru /etc/security/passwd do súboru /home/ftp/etc/security/passwd:
    ```
    cp /etc/security/passwd /home/ftp/etc/security/passwd
    ```
27. Súbor /home/ftp/etc/security/passwd upravte pomocou obľúbeného editora. Napríklad:
    ```
    vi ./security/passwd
    ```
28. Zo skopírovaného obsahu odstráňte všetky sekcie okrem sekcie pre testovacieho používateľa.
29. Zo sekcie testovacieho užívateľa odstráňte riadok flags = ADMCHG. Obsah súboru by sa mal po vykonaní úprav podobať nasledujúcemu:

```
test:
        password = 2HaAYgpDZX3Tw
        lastupdate = 990633278
```

30. Uložte zmeny a ukončite editor.
31. Zadaním nasledujúceho príkazu zmeňte oprávnenia súboru /home/ftp/etc/security/passwd na -rw-------:

    ```
    chmod 600 ./security/passwd
    ```

32. Pomocou nasledujúceho príkazu zmeňte vlastníka a skupinu súboru /home/ftp/etc/security/passwd na užívateľa typu root a skupinu zabezpečenia:

    ```
    chown root:security ./security/passwd
    ```

33. S použitím svojho obľúbeného editora vytvorte a upravte súbor /home/ftp/etc/group. Napríklad:

    ```
    vi group
    ```

34. Do súboru pridajte tieto riadky:

    ```
    system:*:0:
    staff:*:1:test
    ```

35. Uložte zmeny a ukončite editor.
36. Zadaním nasledujúceho príkazu zmeňte oprávnenia súboru /home/ftp/etc/group na -rw-r--r—:

    ```
    chmod 644 group
    ```

37. Pomocou nasledujúceho príkazu zmeňte vlastníka a skupinu súboru /home/ftp/etc/group na užívateľa typu root a skupinu zabezpečenia:

    ```
    chown root:security group
    ```

38. S použitím svojho obľúbeného editora vytvorte a upravte súbor /home/ftp/etc/security/group. Napríklad:

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

39. Do súboru pridajte tieto riadky:

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

40. Uložte zmeny a ukončite editor. Urobte tak pomocou nasledujúcich krokov:
    a. Skopírujte súbor /etc/security/user do adresára /home/ftp/etc/security tak, že zadáte:

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

    b. Odstráňte všetky odseky z kopírovaného obsahu okrem odseku pre užívateľa test. Použite na to editor, pomocou ktorého napíšte:

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

    c. Uložte zmeny a ukončite editor.
41. Zadaním nasledujúceho príkazu zmeňte oprávnenia súboru /home/ftp/etc/security/group na -rw-r-----:

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

42. Pomocou nasledujúceho príkazu zmeňte vlastníka a skupinu súboru /home/ftp/etc/security/group na užívateľa typu root a zabezpečenie:

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

43. Na kopírovanie príslušného obsahu do adresára /home/ftp/etc/objrepos použite tieto príkazy:

    ```
    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. Zadaním nasledujúceho príkazu prejdite do adresára /home/ftp/home:

    ```
    cd ../home
    ```

45. Zadaním nasledujúceho príkazu vytvorte pre užívateľa nový domovský adresár:

```
mkdir test
```

Tento adresár bude domovským adresárom nového užívateľa ftp.

46. Pomocou nasledujúceho adresára zmeňte vlastníka a skupinu adresára /home/ftp/home/test na užívateľa test a skupinu staff:
    ```
    chown test:staff test
    ```
47. Zadaním nasledujúceho príkazu zmeňte oprávnenia súboru /home/ftp/home/test na -rwx------:
    ```
    chmod 700 test
    ```
48. Zakážte vzdialené prihlasovanie a prihlasovanie prostredníctvom konzoly pre užívateľa test zadaním:
    ```
    chuser login=false rlogin=false test
    ```

Vykonaním týchto krokov ste v počítači nastavili podradené prihlásenie. Otestovať ho môžete pomocou tejto procedúry:

1. Pomocou príkazu ftp sa pripojte k hostiteľskému počítaču, na ktorom ste vytvorili užívateľa test. Napríklad:
   ```
   ftp MyHost
   ```
2. Prihláste sa ako užívateľ anonymous. Po vyzvaní na zadanie hesla stlačte kláves Enter.
3. Pomocou nasledujúceho príkazu prepnite na novovytvoreného užívateľa test:
   ```
   user test
   ```

   Po zobrazení výzvy na zadanie hesla zadajte heslo, ktoré ste vytvorili v kroku 18 na strane 56.
4. Pomocou príkazu **pwd** skontrolujte, či existuje domovský adresár užívateľa. Napríklad:
   ```
   ftp> pwd
        /home/test
   ```

   Výstup zobrazuje adresár /home/test ako podadresár pripojenia ftp. Úplná cesta na hostiteľskom počítači je v skutočnosti /home/ftp/home/test.

Poznámky:

- Užívateľov môžete prepnúť len pomocou podriadených užívateľov ftp. Napríklad test je podriadený užívateľ ftp.
- Keď pomocou skriptu anon.users.ftp vytvoríte anonymných užívateľov ftp, užívateľovi môžete priradiť ľubovoľný názov tým, že v skripte nahradíte *username*.
- Keďže server vykonáva príkaz **chroot** v domovskom adresári užívateľského konta, v prípade anonymných užívateľov by sa všetky súbory súvisiace s konfiguráciou, ako napríklad *fileftpaccess.ctl*, mali nachádzať v domovskom adresári, napríklad ~/etc/, príslušného anonymného užívateľa. V prípade obmedzení 'writeonly,' 'readonly,' a 'readwrite,' v súbore /etc/ftpaccess.ctl musí existovať cesta relatívna k ceste zmenenej príkazom chroot.

Bližšie informácie nájdete v:

- "Bezpečnosť TCP/IP" v *Security*
- "**Príkaz ftp** " v *Commands Reference*

Systémové špeciálne užívateľské kontá

AIX poskytuje predvolenú množinu špeciálnych užívateľských kont systému, ktoré zabraňujú, aby kontá typu root a systémové kontá vlastnili všetky súbory operačného systému a súborové systémy.

Upozornenie: Pri odstraňovaní špeciálneho užívateľského konta systému postupujte opatrne. Konkrétne konto môžete zakázať vložením hviezdičky (*) na začiatok jemu zodpovedajúceho riadka v súbore /etc/security/passwd. Pri zakazovaní konta užívateľa s oprávneniami typu root však buďte opatrní. Ak odstránite špeciálne užívateľské kontá systému or alebo zakážete konto typu root, operačný systém sa znefunkční.

V operačnom systéme sú preddefinované nasledovné kontá:

adm Užívateľské konto adm vlastní nasledovné základné systémové funkcie:

- Diagnostiku, ktorej nástroje sú uložené v adresári /usr/sbin/perf/diag_tool.
- Evidenciu, ktorej nástroje sú uložené v nasledujúcich adresároch:
  - /usr/sbin/acct
  - /usr/lib/acct
  - /var/adm
  - /var/adm/acct/fiscal
  - /var/adm/acct/nite
  - /var/adm/acct/sum

bin Užívateľské konto typu bin obvykle vlastní spustiteľné súbory pre väčšinu príkazov užívateľa. Primárnym účelom tohto konta je napomáhať pri distribúcii vlastníctva dôležitých systémových adresárov a súborov, takže všetko nebudú vlastniť len užívateľské kontá typu root a sys.

daemon
Užívateľské konto dymu démon existuje len za účelom vlastnenia a spúšťania procesov servera a súvisiacich súborov. Toto konto zabezpečuje, že tieto procesy sa budú spúšťať s príslušnými oprávneniami na prístup k súborom.

nobody
Užívateľské konto Nikto sa používa v systéme Network File System (NFS) na povolenie vzdialenej tlače. Toto konto existuje, aby program mohol povoliť dočasný prístup typu root k užívateľom s oprávneniami typu root. Napríklad pred povolením Secure RPC a Secure NFS skontrolujte kľúč /etc/public na hlavnom serveri NIS a nájdite užívateľa, ktorému nebol priradený verejný kľúč a tajný kľúč. Ako užívateľ s oprávneniami typu root môžete v databáze vytvoriť položku pre každého nepriradeného užívateľa zadaním:

```
newkey -u meno užívateľa
```

Môžete tiež v databáze vytvoriť položku pre užívateľské konto typu nikto, aby potom každý užívateľ mohol spustiť program **chkey** na vytvorenie vlastných položiek v databáze bez toho, aby sa prihlásil ako užívateľ s oprávneniami typu root.

root Užívateľské konto typu root, UID 0, prostredníctvom ktorého môže vykonávať úlohy údržby systému a odstraňovať problémy so systémom.

sys Užívateľ sys vlastní štandardný bod pripojenia pre pamäť cache DFS (Distributed File Service), ktorá musí existovať pred začatím inštalácie alebo konfigurácie DFS na klientovi. Adresár /usr/sys tiež môže uschovávať inštalačné obrazy.

system Systémová skupina je systémom definovaná skupina správcov systému. Používatelia, patriaci do tejto systémovej skupiny, majú oprávnenie plniť niektoré úlohy údržby systému bez toho, aby museli mať oprávnenia hlavného používateľa (root).

Odstránenie nepotrebných štandardných užívateľských kont:

Počas inštalácie operačného systému sa vytvára množstvo predvolených ID užívateľov a skupín. V závislosti od aplikácií, ktoré v systéme spúšťate a od umiestnenia systému v sieti, môžu niektoré z týchto ID užívateľov a skupín predstavovať slabé miesta zabezpečenia, ktoré možno zneužiť.

Nasledovná tabuľka obsahuje zoznam najčastejších predvolených ID užívateľov, ktoré je obvykle možné odstrániť:

Tabuľka 3. Najčastejšie predvolené ID užívateľov, ktoré je obvykle možné odstrániť.

ID užívateľa	Popis
uucp, nuucp	Vlastník skrytých súborov používaných protokolom uucp. Užívateľské konto uucp sa používa pre UNIX-to-UNIX Copy Program, ktorý je skupinou príkazov, programov a súborov, ktoré sa nachádzajú na väčšine systémov AIX, ktoré užívateľom umožňujú komunikovať s ďalším systémom AIX cez vyhradenú linku alebo cez telefónnu linku.
lpd	Vlastník súborov používaných subsystémom tlače
guest	Umožňuje prístup k užívateľom, ktorí nemajú prístup ku kontám

Nasledovná tabuľka obsahuje zoznam najčastejších ID skupín, ktoré obvykle nie sú potrebné:

Tabuľka 4. Najčastejšie ID skupín, ktoré obvykle nie sú potrebné.

ID skupiny	Popis
uucp	Skupina, do ktorej patria užívatelia uucp a nuucp
printq	Skupina, do ktorej patrí užívateľ lpd

Zanalyzujte svoj systém, aby ste zistili, ktoré ID skutočne nie sú potrebné. Nepotrebné môžu byť aj ďalšie ID užívateľov a skupín. Pred uvedením systému do prevádzky starostlivo zvážte použitie dostupných ID.

Poznámka: Miesto odstránenia skupiny printq v dôsledku závislosti na sadách súborov tlačiarne zakážte identifikátor užívateľa lp, príkaz **piobe** a program démona qdaemon v položke súboru /etc/inittab v záujme minimalizácie bezpečnostných rizík. Zabránite tak užívateľovi v spúšťaní príkazov **print**.

Kontá vytvorené bezpečnostnými komponentmi:

Pri inštalácii alebo konfigurácii bezpečnostných komponentov, napríklad LDAP a OpenSSH, sa vytvoria užívateľské a skupinové kontá.

Vytvorené užívateľské a skupinové kontá zahŕňajú:

- **Bezpečnosť internetového protokolu (IP):** Bezpečnosť IP pridáva užívateľský *ipsec* a skupinový *ipsec* počas inštalácie. Tieto ID používa služba riadenia kľúčov. Všimnite si, že ID skupiny v /usr/lpp/group.id.keymgt nie je možné prispôsobiť pred inštaláciou.
- **Kerberos a Public Key Infrastructure (PKI):** Tieto komponenty nevytvárajú žiadne nové užívateľské alebo skupinové kontá.
- **LDAP:** Keď sa inštaluje server alebo klient LDAP, vytvorí sa užívateľské konto *ldap*. ID užívateľa *ldap* nie je opravené. Počas inštalácie servera LDAP sa automaticky nainštaluje databáza DB2. Inštalácia DB2 vytvorí skupinové konto *dbsysadm*. Predvolený ID skupiny *dbsysadm* je 400. Príkaz **mksecldap** vytvorí počas konfigurácie servera LDAP užívateľské konto *ldapdb2*.
- **OpenSSH:** Počas inštalácie OpenSSH bude do systému pridaný užívateľský *sshd* a skupinový *sshd*. Príslušné ID užívateľa a skupiny sa nesmú meniť. Funkcia separácie privilégií v SSH vyžaduje ID.

Skupiny bez domén

Funkcia skupín bez domén vám umožňuje zaradiť užívateľov definovaných v jednej doméne do skupín definovaných v inej doméne. Táto funkcia podporuje iba domény LDAP (Lightweight Database Access Protocol) a lokálne domény.

Pomocou modulu LDAP Authentication Load Module (modul LDAP) môžete vytvárať užívateľov a skupiny na serveri LDAP. Taktiež môžete pomocou modulu Local Authentication Load Module (lokálny modul) vytvárať užívateľov a skupiny v lokálnom systéme. Ak funkcia **domainlessgroups** nie je povolená, užívateľov a skupiny, ktoré neboli vytvorené v adresári LDAP alebo v lokálnom systéme, nie je možné zaradiť do skupín mimo domény, v ktorej boli vytvorené. Napríklad, užívateľa, ktorý bol vytvorený v doméne LDAP, nie je možné zaradiť do skupiny v lokálnej doméne.

Povolením systémovej vlastnosti **domainlessgroups** môžete obísť toto obmedzenie a zaradiť užívateľov do skupín v adresári LDAP aj do lokálnych skupín. Vlastnosť **domainlessgroups** je definovaná v súbore /etc/secvars.cfg. Táto funkcia je však podporovaná iba pre moduly LDAP a lokálne moduly. Možné hodnoty tejto vlastnosti sú:

false (predvolená hodnota)
Atribút skupiny sa nezlučuje z modulov LDAP a lokálnych modulov.

true Atribút skupiny sa zlučuje z modulov LDAP a lokálnych modulov. Napríklad, užívateľov z adresára LDAP môžete zaradiť do lokálnych skupín.

Hodnotu vlastnosti **domainlessgroups** môžete zistiť pomocou príkazu:

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

Ak chcete vlastnosť **domainlessgroups** nastaviť na hodnotu true, zadajte nasledujúci príkaz:

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

Nasledujúca tabuľka uvádza rozdiely vo výsledkoch príkazov pre užívateľov a skupiny v závislosti od nastavenia vlastnosti **domainlessgroups**.

Tabuľka 5. Výsledky vybratých príkazov, na ktoré má vplyv vlastnosť **domainlessgroups**

Príkaz	Výsledok, keď je vlastnosť domainlessgroups nastavená na hodnotu true
chgroup -R **ldap**|**files**	Aktualizuje skupinu v určenej doméne. Užívateľa môžete pridať do skupiny v adresári LDAP alebo lokálnej skupiny.
chuser -R **ldap**|**files**	Zmení nastavenia pre užívateľa v určenej doméne. Ak sú určené skupiny, ktoré sú definované v druhej doméne, v týchto skupinách sa tiež premietnu informácie o užívateľovi.
login *meno užívateľa* alebo su	Získa užívateľské atribúty z registra užívateľov okrem atribútu identifikátora skupiny. Užívateľské atribúty pre identifikátor skupiny sa zlúčia z domény LDAP aj z lokálnej domény.
lsgroup -R **ldap**|**files**	Zobrazí zoznam všetkých skupinových atribútov pre vybratú doménu. Ak nenájde určenú skupinu v uvedenej doméne, príkaz zlyhá.
lsuser -R **ldap**|**files**	Zobrazí atribúty užívateľa po zlúčení týchto informácií zo všetkých skupín v doméne, v ktorých je tento užívateľ definovaný a druhej domény. Ak primárna skupina užívateľa nie je definovaná v doméne, v ktorej je definovaný užívateľ, prevezme sa z druhej domény.
mkgroup -R **ldap**|**files**	Vytvorí skupinu v určenej doméne. Po vytvorení skupiny musíte užívateľa (či už užívateľa LDAP alebo lokálneho užívateľa) zaradiť do skupiny v databáze skupín danej domény. Užívateľa môžete pridať do skupín v adresári LDAP alebo lokálnych skupín.
mkuser -R **ldap**|**files**	Vytvorí užívateľa v určenej doméne. Ak sú určené skupiny, ktoré sú definované v druhej doméne, v týchto skupinách sa tiež premietnu informácie o užívateľovi.
rmgroup -R **ldap**|**files**	Odstráni uvedenú skupinu z určenej domény. Ak je skupina určená ako primárna skupina užívateľa, ktorý je definovaný v niektorej doméne, príkaz zlyhá.
rmuser -R **ldap**|**files**	Odstráni uvedeného užívateľa z určenej domény. Taktiež odstráni užívateľa zo všetkých skupín, ktoré sú definované v druhej doméne a obsahujú tohto užívateľa ako člena.

Súvisiace koncepty:

“Zavádzací modul autentifikácie LDAP” na strane 146
Bezpečnostný subsystém v rámci LDAP je implementovaný ako zavádzací modul autentifikácie LDAP. Koncepčne sa podobá ostatným zavádzacím modulom, napríklad NIS, DCE a KRB5. Zavádzacie moduly sú definované v súbore /usr/lib/security/methods.cfg.

Súvisiace informácie:

príkaz chgroup

príkaz chuser

príkaz login

príkaz lsgroup

príkaz lsuser

príkaz mkgroup

príkaz mkuser

príkaz rmgroup

príkaz rmuser

príkaz su

Heslá

Najčastejšie používanou metódou narušenia za účelom získania prístupu do systému je hádanie hesla. Preto riadenie a monitorovanie vašej politiky obmedzení hesiel je nevyhnutné.

Systém AIX poskytuje mechanizmy, ktoré pomáhajú zaistiť bezpečnejšiu politiku hesla, napríklad zavedenie nasledovných hodnôt:

- Minimálny a maximálny počet týždňov pred a po dátume, kedy je možné zmeniť heslo
- Minimálna dĺžka hesla
- Minimálny počet abecedných znakov, ktoré je možné použiť pri výbere hesla

Vytvorenie dobrých hesiel:

Dobré heslá sú účinné prvé obranné línie proti neoprávnenému vstupu do systému.

Heslá sú účinné, ak:

- Obsahujú veľké aj malé písmená
- Predstavujú kombináciu abecedných znakov, čísel a interpunkčných znamienok a môžu obsahovať aj špeciálne znaky, ako napríklad ~!@#$%^&*()-_=+[]{}|\;:'",.<>?/<medzeru>
- Nie sú nikde zapísané
- Majú dĺžku minimálne 7 a maximálne 8 znakov, pri použití súboru /etc/security/passwd (autentifikačná implementácia, ktorá používa registre, napríklad LDAP, môžu heslá prekročiť túto maximálnu dĺžku)
- Nepredstavujú skutočné slová, ktoré možno nájsť v každom slovníku
- Nepredstavujú vzory nastavenia písma na klávesnici, napríklad *qwerty*
- Nepredstavujú odzadu napísané reálne slová alebo známe vzory
- Neobsahujú žiadne osobné informácie o vašej osobe, rodine alebo priateľoch
- Nie sú tvorené podľa rovnakého vzoru ako predchádzajúce heslo
- Heslo možno napísať relatívne rýchlo, aby ho nikto v okolí nemohol odpozorovať

Okrem týchto mechanizmov si môžete vynútiť prísnejšie pravidlá, keď uplatníte obmedzenie, že heslá nemôžu obsahovať štandardné slová systému UNIX, ktoré sa dajú uhádnuť. Táto funkcia používa dictionlist, ktorý vyžaduje, aby ste si najprv nainštalovali sady súborov bos.data a bos.txt.

Ak chcete implementovať predtým definovaný dictionlist, v súbore /etc/security/users upravte nasledujúci riadok:

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

Súbor /usr/share/dict/words používa dictionlist, aby zamedzil používaniu štandardných slov systému UNIX ako hesiel.

Používanie súboru /etc/passwd:

Súbor /etc/passwd sa už tradične používa na sledovanie každého registrovaného užívateľa, ktorý má prístup do systému.

Tento súbor /etc/passwd je oddelený dvojbodkami a obsahuje nasledovné informácie:

- Meno užívateľa
- Zašifrované heslo
- Číslo ID užívateľa (UID)
- ID číslo užívateľskej skupiny (GID)
- Celé meno užívateľa (GECOS)
- Domovský adresár užívateľa
- Prostredie pre prihlasovanie

Nasleduje príklad súboru /etc/passwd:

```
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
```

AIX neukladá šifrované heslá v súbore /etc/password ako systémy UNIX, ale ukladá ich štandardne do súboru /etc/security/password [1], ktorý môže čítať len užívateľ root. Heslo zapísané do súboru /etc/passwd používa systém AIX na zistenie, či dané heslo existuje, alebo či je konto blokované.

Vlastníkom súboru /etc/passwd je užívateľ s oprávneniami typu root a súbor musí byť k dispozícii na čítanie pre všetkých užívateľov. Oprávnenia na zápis má však len užívateľ s oprávneniami typu root, čo je zobrazené ako -rw-r--r--. Ak ID užívateľa má heslo, pole hesla bude obsahovať ! (výkričník). Ak ID užívateľa nemá heslo, pole hesla bude obsahovať * (hviezdičku). Šifrované heslá sú uložené v súbore /etc/security/passwd. Nasledujúci príklad obsahuje posledné štyri položky v súbore /etc/security/passwd podľa položiek z vyššie uvedeného súboru /etc/passwd.

```
guest:
        password = *

nobody:
        password = *

lpd:
        password = *

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

ID užívateľa jdoe nemá položku v súbore /etc/security/passwd, lebo nemá nastavené heslo v súbore /etc/passwd.

Konzistenciu súboru /etc/passwd možno skontrolovať použitím príkazu **pwdck**. Príkaz **pwdck** overí správnosť informácií o hesle v databázových súboroch užívateľov kontrolou definícií pre všetkých alebo len pre konkrétnych užívateľov.

1. /etc/security/password

Používanie súboru /etc/passwd a sieťových prostredí:

V tradičnom sieťovom prostredí užívateľ musel mať konto na každom systéme, aby mohol získať prístup do toho systému.

Prakticky to znamená, že užívateľ by mal prístup do všetkých súborov /etc/passwd v každom systéme. Ale v distribuovanom prostredí neexistuje jednoduchý spôsob na zabezpečenie, aby každý systém mal rovnaký súbor /etc/passwd. Aby sa vyriešil tento problém, informácie v súbore /etc/passwd sa cez sieť niekoľkými spôsobmi vrátane systému Network Information System (NIS).

Ukrývanie užívateľských mien a hesiel:

Ak chcete dosiahnuť vyššiu úroveň zabezpečenia, zabezpečte, aby ID a heslá neboli v rámci systému viditeľné.

Súbory .netrc obsahujú ID a heslo užívateľa. Tento súbor nie je chránený šifrovaním ani kódovaním a jeho obsah je zobrazený ako obyčajný text. Na vyhľadanie týchto súborov zadajte nasledovný príkaz:

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

Vyhľadané súbory potom odstráňte. Efektívnejší spôsob uloženia hesiel je nastavenie systému Kerberos. Ďalšie informácie o protokole Kerberos nájdete v časti “Kerberos” na strane 274.

Nastavenie odporúčaných volieb pre heslá:

Optimálnu správu hesiel možno dosiahnuť len prostredníctvom vzdelávania užívateľov. Na zabezpečenie dodatočnej bezpečnosti poskytuje operačný systém konfigurovateľné obmedzenia pre heslá. Tieto administrátorovi umožňujú obmedziť heslá, ktoré si zvolili užívatelia a vynútiť si pravidelnú zmenu hesiel.

Voľby pre heslá a rozšírené atribúty užívateľov sa nachádzajú v súbore /etc/security/user, súbore ASCII, ktorý obsahuje sekcie atribútov pre užívateľov. Tieto obmedzenia sú vynútené pri každom definovaní nového hesla pre užívateľa. Všetky obmedzenia hesiel sú definované pre konkrétneho užívateľa. Keďže pre obmedzenia sa používajú predvolené sekcie v súbore /etc/security/user, výsledkom vynútenia obmedzení pre všetkých užívateľov sú rovnaké obmedzenia. Ak chcete zachovať bezpečnosť hesiel, všetky heslá musia mať podobnú ochranu.

Administrátori môžu tiež rozširovať obmedzenia pre heslá. Použitím atribútu **pwdchecks** v súbore /etc/security/user môže administrátor do kódu obmedzení hesiel pridať nové subrutiny (známe ako *metódy*). Takže operačný systém môže pridať a vynútiť lokálne politiky. Viac informácií nájdete v časti “Rozšírenie obmedzení hesiel” na strane 68.

Obmedzenia hesiel používajte uvážene. Pokusy o striktné obmedzenia, napríklad obmedzenie dĺžky hesla, čo uľahčuje uhádnutie hesla, alebo vynútený výber hesla, ktoré sa ťažko pamätá a užívateľ si ho potom možno zapíše, môžu ohroziť bezpečnosť hesla. Bezpečnosť hesla je nakoniec v rukách užívateľa. Jednoduché obmedzenia pre heslá spojené s rozumnými smernicami a občasným auditom, ktorý overí jedinečnosť aktuálnych hesiel, sú najlepšou politikou.

V nasledovnej tabuľke sú uvedené odporúčané hodnoty pre niektoré atribúty zabezpečenia vzťahujúce sa na heslá užívateľov v súbore /etc/security/user.

Tabuľka 6. Odporúčané hodnoty atribútov zabezpečenia pre heslá užívateľov.

Atribút	Popis	Odporúčaná hodnota	Predvolená hodnota	Maximálna hodnota
dictionlist	Overuje, či heslá neobsahujú štandardné slová systému UNIX.	**/usr/share/dict/words**	Nepoužiteľné	Nepoužiteľné
histexpire	Počet týždňov pred opätovným použitím hesla.	26	0	260*
histsize	Povolený počet iterácií hesla.	20	0	50

Tabuľka 6. Odporúčané hodnoty atribútov zabezpečenia pre heslá užívateľov. (pokračovanie)

Atribút	Popis	Odporúčaná hodnota	Predvolená hodnota	Maximálna hodnota
maxage	Maximálny počet týždňov pred zmenou hesla.	8	0	52
maxexpired	Maximálny počet týždňov po období *maxage,* kedy môže užívateľ zmeniť heslo so skončenou platnosťou. (Užívateľ s oprávneniami typu root je od toho oslobodený.)	2	-1	52
maxrepeats	Maximálny počet znakov, ktoré sa môžu v hesle opakovať.	2	8	8
minage	Minimálny počet týždňov pred dátumom, kedy je možné zmeniť heslo. Nemal by sa nastavovať na nenulovú hodnotu, iba ak sú administrátori vždy ľahko dosiahnuteľní, aby mohli nastaviť náhodne prezradené heslo, ktoré bolo nedávno zmenené.	0	0	52
minalpha	Minimálny počet abecedných znakov požadovaných pre heslo.	2	0	PW_PASSLEN**
mindiff	Minimálny počet jedinečných znakov, ktoré musí heslo obsahovať.	4	0	PW_PASSLEN**
minlen	Minimálna dĺžka hesla.	6 (8 pre užívateľa s oprávneniami typu root)	0	PW_PASSLEN**
minother	Minimálny počet iných než abecedných znakov požadovaných pre heslo.	2	0	PW_PASSLEN**
pwdwarntime	Počet dní pred zobrazením varovania, že zmena hesla sa požaduje.	5	Nepoužiteľné	Nepoužiteľné
pwdchecks	Túto položku možno použiť na rozšírenie príkazu **passwd** vlastným kódom na kontrolu kvality hesla.	Viac informácií nájdete v časti “Rozšírenie obmedzení hesiel” na strane 68.	Nepoužiteľné	Nepoužiteľné

* Maximum je 50 uchovávaných hesiel.

** PW_PASSLEN je definovaný v súbore userpw.h.

Ak je v systéme nainštalované spracovanie textu, administrátor dokáže súbor /usr/share/dict/words použiť ako slovníkový súbor **dictionlist**. V tomto prípade môže administrátor nastaviť atribút **minother** na hodnotu 0. Keďže väčšina slov v súbore slovníka neobsahuje znaky spadajúce do kategórie atribútov **minother**, nastavením atribútu **minother** na hodnotu 1 alebo vyššiu vylúčime použitie väčšiny slov v tomto slovníku.

Minimálna dĺžka hesla v systéme je nastavená hodnotou atribútu **minlen** alebo súčtom hodnôt atribútu **minalpha** a atribútu **minother** (použitá je vyššia hodnota).

Maximálna dĺžka hesla je počet znakov určený atribútom **PW_PASSLEN**. Počet znakov použitých pri generovaní uloženej hodnoty hesla je závislý od algoritmu hesiel, ktorý systém používa. Algoritmy hesla sú definované v súbore /etc/security/pwdalg.cfg a predvolený algoritmus hesla, ktorý bude použitý, môže byť nakonfigurovaný prostredníctvom atribútu **pwd_algorithm** v súbore /etc/security/login.cfg. Súčet hodnôt atribútu **minalpha** a atribútu **minother** nesmie byť nikdy vyšší, než atribút **PW_PASSLEN**. Ak je súčet hodnôt atribútu **minalpha** a atribútu **minother** vyšší než atribút **PW_PASSLEN**, bude hodnota atribútu **minother** znížená na hodnotu rozdielu medzi atribútom **PW_PASSLEN** a atribútom **minalpha**.

Ak sa nastavia hodnoty atribútov **histexpire** a **histsize**, systém uschová taký počet hesiel, aký je potrebný na splnenie oboch podmienok, až po ohraničenie 50 hesiel na jedného užívateľa. Heslá s hodnotou Null sa neuchovajú.

Súbor /etc/security/user môžete upraviť tak, aby obsahoval všetky štandardné hodnoty, ktoré chcete použiť na správu užívateľských hesiel. Ďalšia možnosť je zmeniť hodnoty atribútov pomocou príkazu **chuser**.

Ostatné príkazy, ktoré možno použiť s týmto súborom, sú **mkuser**, **lsuser** a **rmuser**. Príkaz **mkuser** vytvorí v súbore /etc/security/user položku pre každého nového užívateľa a inicializuje jeho atribúty pomocou atribútov, ktoré sú definované v súbore /usr/lib/security/mkuser.default. Atribúty a ich hodnoty zobrazíte príkazom **lsuser**. Na odstránenie užívateľa použite príkaz **rmuser**.

Podpora hesiel s viac ako 8 znakmi a algoritmus LPA (Loadable Password Algorithm):

V dôsledku najnovších pokrokov v oblasti počítačového hardvéru je tradičné šifrovanie hesiel v systéme UNIX málo bezpečné pri útokoch založených na hrubom hádaní hesiel. Z kryptografického hľadiska slabý algoritmus môže odhaliť dokonca aj silné heslá. Systém AIX podporuje algoritmus LPA (Loadable Password Algorithm), ktorý poskytuje bezpečné mechanizmy na šifrovanie hesiel.

Funkcia tradičného šifrovania hesiel:

Štandardný mechanizmus autentifikácie systému AIX používa na autentifikáciu užívateľov jednosmernú hašovaciu funkciu s názvom **crypt**. Funkcia **crypt** je modifikovaný algoritmus DES, ktorý vykonáva jednosmerné šifrovanie poľa pevných údajov pomocou poskytnutého hesla a závislej hodnoty šifrovania.

Funkcia **crypt** použije len prvých osem znakov z reťazca hesla; heslo užívateľa sa skráti na osem znakov. Ak heslo obsahuje menej než osem znakov, bude vyplnené nulovými bitmi na pravej strane. 56-bitový kľúč DES je odvodený na základe 7 bitov z každého znaku.

Závislá hodnota šifrovania je 2-znakový reťazec (12 bitov závislej hodnoty šifrovania sa použijú na rušenie algoritmu DES) zo znakovej sady "**A**-**Z**", "**a**-**z**","**0**-**9**","**.**" (bodka) a "/". Závislá hodnota šifrovania sa používa na zmenu hašovacieho algoritmu, takže rovnaké heslo čistého textu môže vytvoriť 4 096 možných zašifrovaní hesla. Môže sa to dosiahnuť úpravou algoritmu DES, kde sa bity i a i+24 vo výstupe DES E-Box vymenia, keď je bit i nastavený v závislej hodnote šifrovania, v dôsledku čoho sa zároveň šifrovací hardvér DES nepoužiteľným pri snahe uhádnuť heslo.

64-bitový blok všetkých nulových bitov je zašifrovaný pomocou kľúča DES 25-krát. Konečným výstupom je 12-bitová závislá hodnota šifrovania zreťazený so zašifrovanou 64-bitovou hodnotou. Výsledná 76-bitová hodnota je zapísaná do 13 vytlačiteľných znakov ASCII v tvare base64.

Hašový algoritmus pre heslá:

Prelomiť hašovacie algoritmy, ako napríklad MD5, je ťažšie než prelomiť funkciu **crypt**. Poskytujú mechanizmus odolný voči hrubej sile útokov, snažiacich sa odhaliť heslo. Keď sa na vytvorenie hašu používa celé heslo, v prípade šifrovania hesla cez hašovací algoritmus neexistuje obmedzenie pre dĺžku hesla.

Loadable Password Algorithm:

Operačný systém AIX 6.1 a novšie verzie implementujú mechanizmus Loadable Password Algorithm (LPA), prostredníctvom ktorého je možné ľahko nasadiť nové algoritmy šifrovania hesiel.

Každý podporovaný algoritmus na šifrovanie hesiel je implementovaný ako zavádzací modul LPA, ktorý sa zavedie v čase runtime, keď je algoritmus potrebný. Podporované algoritmy LPA a ich atribúty sú zadefinované v súbore konfigurácie systému /etc/security/pwdalg.cfg.

Administrátor môže nastaviť celosystémový mechanizmus šifrovania hesiel, ktorý na šifrovanie hesiel používa špecifický algoritmus LPA. Keď sa celosystémový mechanizmus hesiel zmení, heslá šifrované pomocou predtým vybratého mechanizmu šifrovania hesiel (napríklad pomocou funkcie **crypt**), budú naďalej podporované.

Podpora pre viac ako osemznakové heslá:

Všetky LPA implementované pre AIX 6.1 a novšie podporujú heslá dlhšie ako osem znakov. Obmedzenia dĺžky hesla sú v rôznych algoritmoch LPA odlišné. Maximálna podporovaná dĺžka hesla je 255 znakov.

Konfiguračný súbor LPA:

Konfiguračný súbor pre LPA je /etc/security/pwdalg.cfg. Je to súbor s odsekmi, ktorý definuje atribúty podporovaných LPA.

V konfiguračnom súbore sú zadefinované nasledujúce atribúty LPA:

- Cesta k modulu LPA
- Voliteľné príznaky, ktoré sú runtime odovzdané modulu LPA

Atribúty LPA, zadefinované v konfiguračnom súbore, sú prístupné cez rozhrania **getconfattr** a **setconfattr**.

Nasledujúci ukážkový odsek v súbore /etc/security/pwdalg.cfg definuje algoritmus LPA s názvom **ssha256**:

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

Algoritmus hesla systému:

Administrátor systému môže nastaviť algoritmus celosystémového hesla zvolením LPA ako algoritmu hašovania hesla. V jednom momente môže existovať len jeden aktívny algoritmus systémového hesla. Algoritmus systémového hesla je definovaný systémovým atribútom **pwd_algorithm** v strofe **usw** v súbore /etc/security/login.cfg.

Platné hodnoty pre atribút **pwd_algorithm** v súbore /etc/security/login.cfg sú názvy strof LPA, ktoré sú definované v súbore /etc/security/pwdalg.cfg. Ďalšou platnou hodnotou pre atribút **pwd_algorithm** je **crypt**, ktorá sa vzťahuje na tradičné šifrovanie **crypt**. Ak sa atribút **pwd_algorithm** vynechá z konfiguračného súboru, ako predvolená hodnota sa použije **crypt**.

Nasledujúci príklad súboru /etc/security/login.cfg používa LPA **ssha256** ako algoritmus šifrovania celosystémového hesla.

```
 ... ...
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
 ... ...
```

Algoritmus systémového hesla nadobudne platnosť len pre novovytvorené heslá a zmenené heslá. Po migrácii všetky následné nové heslá alebo zmeny hesiel používajú algoritmus systémového hesla. Heslá, ktoré existovali pred zvolením algoritmu systémového hesla, vygenerované či už štandardnou funkciou **crypt** alebo inými podporovanými modulmi LPA, stále budú v systéme fungovať. Preto zmiešané heslá, ktoré boli vygenerované inými LPA, nemôžu v systéme koexistovať.

Nastavenie algoritmu hesla systému:

Administrátor systému môže pomocou príkazu **chsec** zadať algoritmus hesla systému alebo pomocou editora, napríklad **vi** manuálne upraviť atribút **pwd_algorithm** v súbore /etc/security/login.cfg.

Odporúča sa zadať algoritmus hesla systému pomocou príkazu **chsec**, keďže príkaz **chsec** automaticky skontrolujte definíciu špecifikovaného LPA.

Použitie príkazu chsec

Spustite nasledujúci príkaz na zadanie **smd5** LPA ako šifrovacieho modulu celosystémového hesla:

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

Keď pomocou príkazu **chsec** upravíte atribút **pwd_algorithm**, príkaz **chsec** skontrolujte súbor /etc/security/pwdalg.cfg, či neobsahuje špecifikovaný LPA. Ak zlyhá táto kontrola, príkaz **chsec** zlyhá.

Použitie editora

Ak pomocou editora manuálne zmeníte hodnotu atribútu **pwd_algorithm** v súbore /etc/security/login.cfg, zaistite, aby špecifikovaná hodnota mala názov strofy, ktorá je zadefinovaná v súbore /etc/security/pwdalg.cfg.

Rozšírenie obmedzení hesiel:

Pravidlá používané programom hesiel pre akceptovanie alebo zamietnutie hesiel (obmedzenia tvorby hesiel) môžu byť správcami systémov rozšírené tak, aby poskytovali špecifické obmedzenia.

Obmedzenia sa rozširujú pridávaním metód, ktoré sa volajú pri zmene hesla. Atribút **pwdchecks** v súbore /etc/security/user určuje volané metódy.

Počnúc vydaním *AIX Version 6.1 Technical Reference*, systém obsahuje popis metódy **metóda_pwdrestrict**, rozhrania podrutín, ktorému musia vyhovovať určené metódy obmedzenia hesiel. Ak chcete správne rozširovať obmedzenia tvorby hesiel, správca systému musí naprogramovať toto rozhranie pri písaní metódy na obmedzovanie hesiel. Pri rozširovaní obmedzení tvorby hesiel postupujte opatrne. Tieto rozšírenia majú priamy vplyv na príkazy **login**, **passwd**, **su** a iné programy. Škodlivý alebo chybný kód môže jednoducho narušiť bezpečnosť systému.

Autentifikácia užívateľov

Identifikácia a autentifikácia sa používajú na vytvorenie užívateľskej identity.

Každý užívateľ sa musí do systému prihlásiť. Používateľ sám určuje meno používateľa konta a heslo, ak konto nejaké vyžaduje (v bezpečnom systéme musia mať všetky kontá nejaké heslo, v opačnom prípade sú zrušené). Ak je heslo správne, užívateľ sa prihlási k danému kontu, pričom získa prístupové práva a privilégiá daného konta. Heslá sú uložené v súboroch /etc/passwd a /etc/security/passwd.

Štandardne sú používatelia definovaní v registri súborov. To znamená, že používateľské konto a informácie o skupine sú uložené v súboroch s jednoduchou znakovou sadou ASCII. Po uvedení zásuvných modulov na zavedenie je možné používateľov definovať aj v ďalších registroch. Napríklad, ak sa administráciu používateľov použije zásuvný modul LDAP, potom sa definície používateľov ukladajú v archíve LDAP. V tomto prípade nebude v súbore /etc/security/user žiadna položka pre užívateľov (pri užívateľských atribútoch **SYSTEM** a **registry** existuje výnimka). Keď sa na administráciu používateľov použije zložený modul na zavedenie (t.j. modul na zavedenie s autentifikačnou a databázovou časťou), databázová polovica určuje, ako sa budú administrovať informácie o konte používateľa AIX a autentifikačná polovica popisuje administráciu, týkajúcu sa autentifikácie a hesla. Použitím istých rozhraní modulov na zavedenie (newuser, getentry, putentry atď.) môže autentifikačná polovica popisovať aj atribúty administrácie používateľského konta charakteristické pre autentifikáciu.

Táto metóda autentifikácie je riadená atribútmi **SYSTEM** a registra, ktoré sú definované v súbore /etc/security/súbor užívateľa. Administrátor systému môže definovať atribút authcontroldomain ako súbor /etc/security/login.cfg, aby

sa atribúty **SYSTEM** a registra získavali z authcontroldomain. Napríklad, authcontroldomain=LDAP spôsobí, že systém bude vyhľadávať atribúty **SYSTEM** a registra užívateľa v adresári LDAP, aby určil autentifikačnú metódu použitú pre tohto užívateľa. V prípade lokálne definovaných užívateľov sa uplatňuje výnimka, pričom nastavenie authcontroldomain sa ignoruje a atribúty **SYSTEM** a registra sa vždy získavajú zo súboru /etc/security/user.

Akceptovateľnými symbolmi pre atribút authcontroldomain sú súbory alebo názvy odsekov súboru /usr/lib/security/methods.cfg.

Hodnota atribútu **SYSTEM** sa definuje prostredníctvom gramatického pravidla. Pomocou tohto gramatického pravidla môžu správcovia systému na autentifikovanie konkrétneho používateľa do daného systému kombinovať jednu alebo viacero metód. Dobre známou metódou sú tokeny (symboly) compat, DCE, files a NONE.

Štandardom systému je compat. Predvolená hodnota SYSTEM=compat systému povie, že má na autentifikáciu použiť lokálnu databázu a v prípade, že nenájde riešenie, má sa obrátiť na databázu Network Information Services (NIS). Token files udáva, že počas autentifikácie sa majú použiť iba lokálne súbory, zatia ľčo SYSTEM=DCE povedie k autenfitikačnému toku DCE.

Token NONE vypína autentifikáciu metódy. Ak sa majú vypnúť všetky autentifikácie, v riadkoch SYSTEM a auth1 sekcie užívateľa musí byť uvedený symbol NONE.

Môžete zadať dve alebo viac metód a kombinovať ich s logickými metódami triedy AND a OR. Napríklad, SYSTEM=DCE OR compat indikuje, že používateľ sa môže prihlásiť, ak sa podarí buď DCE alebo lokálna autentifikácia (crypt()), v takto zadanom poradí.

Podobným spôsobom môže správca systému použiť pre atribút **SYSTEM** názvy autentifikačných modulov na zavedenie. Napríklad, keď je atribút **SYSTEM** nastavený na hodnotu SYSTEM=KRB5files ALEBO compat, hostiteľ AIX najprv pre autentifikáciu vyskúša tok Kerberos a ak by zlyhal, až potom vyskúša štandardnú autentifikáciu AIX.

Atribúty **SYSTEM** a **registry** sa vždy uložia do lokálneho súborového systému v súbore /etc/security/user. Ak je užívateľ AIX definovaný v LDAP a atribúty **SYSTEM** a **registry** sú podľa toho nastavené, potom bude mať užívateľ položku v súbore /etc/security/user.

Užívateľské atribúty **SYSTEM** a **registry** sa dajú zmeniť pomocou príkazu **chuser**.

Akceptovateľné tokeny pre atribút **SYSTEM** sa dajú definovať v súbore /usr/lib/security/methods.cfg.

Poznámka: Autentifikácia užívateľa s oprávneniami typu root sa vždy vykonáva použitím súboru zabezpečenia lokálneho systému. Položka atribútu **SYSTEM** pre užívateľa s oprávneniami typu root je v súbore /etc/security/user špecificky nastavená na hodnotu SYSTEM=compat.

Alternatívne metódy autentifikácie boli zapracované do systému pomocou atribútu **SYSTEM**, ktorý sa objavuje v súbore /etc/security/user. Napríklad, prostredie DCE (Distributed Computing Environment) vyžaduje autentifikáciu hesiel, ale platnosť týchto hesiel overuje inak ako model šifrovania, ktorý sa používa v súboroch etc/passwd a /etc/security/passwd. Užívatelia, ktorí sa autentifikujú pomocou DCE môžu mať svoju stať v súbore /etc/security/user nastavenú na SYSTEM=DCE.

Ďalšími hodnotami atribútu **SYSTEM** sú symboly **compat**, **files** a **NONE**. Symbol compat sa používa v prípade, že rozlíšenie názvu (a nasledovná autentifikácia) prehľadáva miestnu databázu, pričom pri negatívnom výsledku sa vykoná hľadanie v databáze NIS (Network Information Services). Symbol files určuje, že pri autentifikácii sa majú použiť len lokálne súbory. Symbol NONE vypína autentifikáciu metód. Ak sa majú vypnúť všetky autentifikácie, v riadkoch SYSTEM a **auth1** sekcie užívateľa musí byť uvedený symbol **NONE**.

Iné akceptovateľné tokeny pre atribút **SYSTEM** sa môžu definovať v súbore /usr/lib/security/methods.cfg.

Poznámka: Autentifikácia užívateľa s oprávneniami typu root sa vždy vykonáva použitím súboru zabezpečenia lokálneho systému. Položka atribútu **SYSTEM** pre užívateľa s oprávneniami typu root je v súbore /etc/security/user špecificky nastavená na hodnotu SYSTEM=compat.

Bližšie informácie o ochrane hesiel nájdete v časti *Operating system and device management*.

Prihlasovacie ID užívateľov

Všetky udalosti auditu zaznamenané pre tohto užívateľa sú označené týmto ID a možno ich overiť pri generovaní záznamov auditu. Bližšie informácie o užívateľských ID prihlásenia nájdete v časti *Operating system and device management*.

Užívateľské a skupinové atribúty podporované zavádzacími modulmi autentifikácie

Sada užívateľských a skupinových atribútov sa používa na dosiahnutie identifikácie a autentifikácie v AIX.

Nasledujúca tabuľka uvádza väčšinu z týchto užívateľských a skupinových atribútov ako zoznam a označuje tiež podporu z rôznych zavádzacích modulov pre tieto atribúty. Každý riadok tabuľky zodpovedá atribútu a každý stĺpec reprezentuje zavádzací modul. Atribúty podporované zavádzacím modulom sú v stĺpci zavádzacieho modulu označené slovom Yes.

Poznámka: PKI a Kerberos sú moduly určené len na autentifikáciu a musia byť kombinované s databázovým modelom (napríklad LOCAL alebo LDAP). Podporujú určité ďalšie (rozšírené) atribúty, iné než atribúty poskytované modelmi LOCAL alebo LDAP. Označenia sa zobrazujú len voči týmto rozšíreným atribútom pre uvedené moduly, aj keď iné atribúty možno funkčne dosiahnuť pomocou modelov LOCAL alebo LDAP.

Tabuľka 7. Užívateľské atribúty a podpora zavádzacieho modulu autentifikácie

Užívateľský atribút	Local	NIS	LDAP	PKI	Kerberos
account_locked	Áno	Nie	Áno	Nie	Nie
admgroups	Áno	Nie	Áno	Nie	Nie
admin	Áno	Nie	Áno	Nie	Nie
auditclasses	Áno	Nie	Áno	Nie	Nie
auth_cert	Nie	Nie	Nie	Áno	Nie
auth_domain	Áno	Nie	Áno	Nie	Nie
auth_name	Áno	Nie	Áno	Nie	Nie
auth1 **Poznámka:** Atribút **auth1** je odmietnutý a už by nemal byť použitý.	Áno	Nie	Áno	Nie	Nie
auth2 **Poznámka:** Atribút **auth2** je odmietnutý a už by nemal byť použitý.	Áno	Nie	Áno	Nie	Nie
capabilities	Áno	Nie	Áno	Nie	Nie
core	Áno	Nie	Áno	Nie	Nie
core_compress	Áno	Nie	Nie	Nie	Nie
core_hard	Áno	Nie	Áno	Nie	Nie
core_naming	Áno	Nie	Nie	Nie	Nie
core_path	Áno	Nie	Nie	Nie	Nie
core_pathname	Áno	Nie	Nie	Nie	Nie
cpu	Áno	Nie	Áno	Nie	Nie
daemon	Áno	Nie	Áno	Nie	Nie
údaje	Áno	Nie	Áno	Nie	Nie
data_hard	Áno	Nie	Áno	Nie	Nie
dce_export	Áno	Nie	Áno	Nie	Nie

Tabuľka 7. Užívateľské atribúty a podpora zavádzacieho modulu autentifikácie (pokračovanie)

Užívateľský atribút	Local	NIS	LDAP	PKI	Kerberos
dictionlist	Áno	Nie	Áno	Nie	Nie
expires	Áno	Nie	Áno	Nie	Áno
príznaky	Áno	Nie	Áno	Nie	Áno
fsize	Áno	Nie	Áno	Nie	Nie
fsize_hard	Áno	Nie	Áno	Nie	Nie
funcmode	Áno	Nie	Áno	Nie	Nie
gecos	Áno	Áno	Áno	Nie	Nie
skupiny	Áno	Áno	Áno	Nie	Nie
groupsids	Áno	Áno	Áno	Nie	Nie
histexpire	Áno	Nie	Áno	Nie	Nie
home	Áno	Áno	Áno	Nie	Nie
host_last_login	Áno	Nie	Áno	Nie	Nie
host_last_unsuccessful_login	Áno	Áno	Áno	Nie	Nie
hostsallowedlogin	Áno	Nie	Áno	Nie	Nie
hostsdeniedlogin	Áno	Nie	Áno	Nie	Nie
id	Áno	Áno	Áno	Nie	Nie
krb5_attributes	Nie	Nie	Nie	Nie	Áno
krb5_kvno	Nie	Nie	Nie	Nie	Áno
krb5_last_pwd_change	Nie	Nie	Nie	Nie	Áno
krb5_max_renewable_life	Nie	Nie	Nie	Nie	Áno
krb5_mknvo	Nie	Nie	Nie	Nie	Áno
krb5_mod_date	Nie	Nie	Nie	Nie	Áno
krb5_mod_name	Nie	Nie	Nie	Nie	Áno
krb5_names	Nie	Nie	Nie	Nie	Áno
krb5_principal	Nie	Nie	Nie	Nie	Áno
krb5_principal_name	Nie	Nie	Nie	Nie	Áno
krb5_realm	Nie	Nie	Nie	Nie	Áno
lastupdate	Áno	Áno	Áno	Nie	Nie
login	Áno	Nie	Áno	Nie	Nie
loginretries	Áno	Nie	Áno	Nie	Nie
logintimes	Áno	Nie	Áno	Nie	Nie
maxage	Áno	Áno	Áno	Nie	Áno
maxexpired	Áno	Áno	Áno	Nie	Nie
maxrepeats	Áno	Nie	Áno	Nie	Nie
maxulogs	Áno	Nie	Áno	Nie	Nie
minage	Áno	Áno	Áno	Nie	Nie
minalpha	Áno	Nie	Áno	Nie	Nie
mindiff	Áno	Nie	Áno	Nie	Nie
mindigit	Áno	Nie	Áno	Nie	Nie
minlen	Áno	Nie	Áno	Nie	Nie
minloweralpha	Áno	Nie	Áno	Nie	Nie
minother	Áno	Nie	Áno	Nie	Nie
minspecialchar	Áno	Nie	Áno	Nie	Nie
minupperalpha	Áno	Nie	Áno	Nie	Nie
nofiles	Áno	Nie	Áno	Nie	Nie

Tabuľka 7. Užívateľské atribúty a podpora zavádzacieho modulu autentifikácie (pokračovanie)

Užívateľský atribút	Local	NIS	LDAP	PKI	Kerberos
nofiles_hard	Áno	Nie	Áno	Nie	Nie
password	Áno	Áno	Áno	Nie	Nie
pgid	Áno	Áno	Nie	Nie	Nie
pgrp	Áno	Áno	Áno	Nie	Nie
projects	Áno	Nie	Áno	Nie	Nie
pwdchecks	Áno	Nie	Áno	Nie	Nie
pwdwarntime	Áno	Nie	Áno	Nie	Nie
rcmds	Áno	Nie	Áno	Nie	Nie
registry	Áno	Nie	Nie	Nie	Nie
rlogin	Áno	Nie	Áno	Nie	Nie
roles	Áno	Nie	Áno	Nie	Nie
rss	Áno	Nie	Áno	Nie	Nie
rss_hard	Áno	Nie	Áno	Nie	Nie
obrazovky	Áno	Nie	Áno	Nie	Nie
shell	Áno	Áno	Áno	Nie	Nie
spassword	Áno	Áno	Áno	Nie	Nie
stack	Áno	Nie	Áno	Nie	Nie
stack_hard	Áno	Nie	Áno	Nie	Nie
su	Áno	Nie	Áno	Nie	Nie
sugroups	Áno	Nie	Áno	Nie	Nie
sysenv	Áno	Nie	Áno	Nie	Nie
SYSTEM	Áno	Nie	Nie	Nie	Nie
time_last_login	Áno	Nie	Áno	Nie	Nie
time_last_unsuccessful_login	Áno	Nie	Áno	Nie	Nie
tpath	Áno	Nie	Áno	Nie	Nie
tty_last_login	Áno	Nie	Áno	Nie	Nie
tty_last_unsuccessful_login	Áno	Nie	Áno	Nie	Nie
ttys	Áno	Nie	Áno	Nie	Nie
umask	Áno	Nie	Áno	Nie	Nie
unsuccessful_login_count	Áno	Nie	Áno	Nie	Nie
unsuccessful_login_times	Áno	Nie	Áno	Nie	Nie
usrenv	Áno	Nie	Áno	Nie	Nie

Tabuľka 8. Skupinové atribúty a podpora zavádzacieho modulu autentifikácie

User attribute	Local	NIS	LDAP	PKI	Kerberos
admin	Áno	Nie	Áno	Nie	Nie
adms	Áno	Nie	Áno	Nie	Nie
dce_export	Áno	Nie	Áno	Nie	Nie
id	Áno	Áno	Áno	Nie	Nie
primary	Áno	Nie	Áno	Nie	Nie
projects	Áno	Nie	Áno	Nie	Nie
obrazovky	Áno	Nie	Áno	Nie	Nie
užívatelia	Áno	Áno	Áno	Nie	Nie

Prehľad systému kvót diskového priestoru

Systémov kvót diskového priestoru umožňuje správcom systému kontrolovať počet súborov a údajových blokov, ktoré možno prideliť užívateľom alebo skupinám.

Koncepcia systému kvót diskového priestoru:

Systém kvót diskového priestoru, ktorý vychádza zo systému Berkeley Disk Quota System, predstavuje účinný spôsob riadenia využívania diskového priestoru. Systém kvót možno definovať pre jednotlivých užívateľov alebo skupiny a udržiava sa pre každý žurnálovaný systém súborov (JFS a JFS2).

Systém kvót disku vytvára obmedzenia založené na nasledujúcich parametroch, ktoré možno zmeniť pomocou príkazu **edquota** pre systémy súborov JFS a príkazu **j2edlimit** pre systémy súborov JFS2:

- Voľné limity pre užívateľa alebo skupinu
- Pevné limity pre užívateľa alebo skupinu
- Doba odkladu pre kvótu

Voľný limit definuje počet diskových blokov s veľkosťou 1 KB alebo počet súborov, ktoré môže užívateľ alebo skupina použiť počas bežných operácií. *Pevný limit* definuje maximálne množstvo diskových blokov alebo súborov, ktoré môže užívateľ kumulovať v rámci stanovených diskových kvót. *Doba odkladu pre kvótu* umožňuje užívateľovi na krátke obdobie prekročiť voľný limit (predvolená hodnota je jeden týždeň). Ak užívateľ do uplynutia zadaného času nezredukuje používané miesto pod hodnotu voľného limitu, systém bude považovať voľný limit za maximálnu možnú veľkosť používaného priestoru a nepridelí užívateľovi žiadny ďalší priestor. Užívateľ môže nastavenie tejto podmienky obnoviť, ak odstránením dostatočného počtu súborov zredukuje používaný priestor pod hodnotu voľného limitu.

Systém kvót diskového priestoru sleduje kvóty užívateľov a skupín v súboroch quota.user a quota.group uložených v hlavných adresároch systémov súborov, ktoré povoľujú používanie kvót. Tieto súbory sa vytvárajú pomocou príkazov **quotacheck** a **edquota** a čítať ich možno pomocou príkazov pre kvóty.

Obnova zo stavu po prekročení kvóty:

Znížením využívania súborového systému môžete vyriešiť podmienky prekročenia kvóty.

Keď ste prekročili limity kvót, na zníženie používania súborového systému môžete použiť nasledujúce metódy:

- Zastavte aktuálny proces, ktorý spôsobil, že súborový systém dosiahol svoju hranicu, odstránením nadbytočných súborov znížte hranicu pod danú kvótu a zopakujte neúspešný program.
- Ak používate editor, ako je napríklad vi, môžete použiť sekvenciu prerušenia daného prostredia na kontrolu miesta používaného súbormi, odstrániť nadbytočné súbory a vrátiť sa bez straty údajov upravovaného súboru. Ak pracujete v prostrediach C alebo Korn, môžete prípadne odstaviť editor klávesovou sekvenciou Ctrl-Z, zadať príkazy pre systém súborov a potom sa vrátiť príkazom **fg** (foreground).
- Dočasne zapíšte súbor do systému súborov, v ktorom limity kvót neboli prekročené, odstráňte nadbytočné súbory a potom vráťte daný súbor do správneho systému súborov.

Nastavenie systému kvót diskového priestoru:

Kvóty diskového priestoru vyžadujú zvyčajne len tie súborové systémy, ktoré obsahujú domovské adresáre a súbory užívateľov.

Použitie systému kvót diskového priestoru zvážte za nasledovných podmienok:

- Diskový priestor systému je obmedzený.
- Vyžadujete vyššiu bezpečnosť systému súborov.
- Úroveň používania disku je vysoká, ako napríklad na mnohých univerzitách.

Ak sa tieto podmienky na vaše prostredie nevzťahujú, asi nebudete chcieť vytvoriť limity používania diskov tým, že implementujete systém kvót diskového priestoru.

Systém kvót diskového priestoru možno používať len so žurnálovaným súborovým systémom.

Poznámka: Nevytvárajte kvóty diskového priestoru pre systém súborov /tmp.

Ak chcete nastaviť systém kvót diskového priestoru, postupujte nasledovne:

1. Prihláste sa s oprávnením užívateľa root.
2. Stanovte, ktoré súborové systémy vyžadujú kvóty.

   Poznámka: Súborový systém /tmp nesmie mať žiadne kvóty, pretože mnohé editory a systémové pomocné programy v ňom vytvárajú dočasné súbory.
3. Príkaz **chfs** použite na zahrnutie konfiguračných atribútov kvót **userquota** a **groupquota** do súboru /etc/filesystems. V nasledujúcom príklade bol príkaz **chfs** použitý na povolenie užívateľských kvót v súborovom systéme /home:

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

   Ak chcete pre systém súborov /home povoliť kvóty užívateľov aj skupín, zadajte:

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

   Zobrazí sa nasledovný zodpovedajúci záznam v súbore /etc/filesystems:

   ```
   /home:
   dev        = /dev/hd1
   vfs        = jfs
   log        = /dev/hd8
   mount      = true
   check      = true
   quota      = userquota,groupquota
   options    = rw
   ```
4. Voliteľné, zadajte náhradné názvy súborov kvót diskového priestoru. Názvy quota.user a quota.group sú predvolené názvy súborov umiestnených v hlavných adresároch systémov súborov s povolenými kvótami. Náhradné názvy alebo adresáre môžete pre tieto súbory kvót zadať pomocou atribútov **userquota** a **groupquota** v súbore /etc/filesystems.

   V nasledujúcom príklade bol príkaz **chfs** použitý na vytvorenie užívateľských a skupinových kvót pre súborový systém /home a na pomenovanie súborov kvót myquota.user a myquota.group:

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

   Zobrazí sa nasledovný zodpovedajúci záznam v súbore /etc/filesystems:

   ```
   /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. Ak neboli zadané súborové systémy predtým pripojené, pripojte ich.
6. Pre každého užívateľa alebo skupinu nastavte požadované limity kvót. Pomocou príkazu **edquota** pre každého užívateľa alebo skupinu vytvorte voľné a pevné limity pre dostupný diskový priestor a maximálny počet súborov.

   Nasledovný vzorový záznam zobrazuje limity kvót pre užívateľa *davec*:

   ```
   Kvóty pre užívateľa davec:
   /home: blocks in use: 30, limits (soft = 100, hard = 150)
                  inodes in use: 73, limits (soft = 200, hard = 250)
   ```

Tento užívateľ použil 30 KB z maximálnej veľkosti 100 KB diskového priestoru. Z maximálneho počtu 200 vytvoril užívateľ *davec* 73 súborov. Tento užívateľ má 50 KB vyrovnávacie pamäte v diskovom priestore a 50 súborov, ktoré sa môžu alokovať do dočasného úložného priestoru.

Keď vytvárate kvóty diskového priestoru pre viacerých užívateľov, na duplikáciu kvót jedného užívateľa inému užívateľovi použite príkaz **edquota** s príznakom **-p**.

Na zdvojenie kvót vytvorených pre užívateľa *davec* pre užívateľa *nanc* zadajte:

```
edquota -p davec nanc
```

7. Systém kvót povolíte pomocou príkazu **quotaon**. Príkaz **quotaon** povolí kvóty pre špecifikovaný súborový systém alebo pre všetky súborové systémy s kvótami (ako sú vyznačené v súbore /etc/filesystems), keď ho použijete s príznakom **-a**.
8. Príkaz **quotacheck** použite na kontrolu konzistentnosti súborov kvót v porovnaní so skutočným využívaním diskového priestoru.

   Poznámka: Urobte to vždy, keď na súborovom systéme po prvýkrát povolíte kvóty a po opätovnom zavedení systému. Vykonanie príkazu **quotacheck** trvá dlhšie na súborovom systéme JFS ako na súborovom systémeJFS2 tej istej veľkosti. Ak sú celú dobu pred opätovným zavedením povolené kvóty, nie je potrebné spúšťať príkaz **quotacheck** na súborovom systéme počas opätovného zavedenia.

   Ak chcete povoliť túto kontrolu a zapnúť kvóty počas spustenia systému, pridajte nasledovné riadky na koniec súboru /etc/rc:

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

Povolený počet skupín

Pre AIX 7.1 môžete nakonfigurovať a opakovane získať povolenú hodnotu Number of Groups. Definuje počet skupín, v ktorých môžu byť užívatelia členmi.

Predvolená hodnota pre Number of Groups je 128. Môžete použiť hodnotu z rozsahu od 128 do 2048. Povolený počet skupín je zadaný v systémovom konfiguračnom parametri v_ngroups_allowed pre zariadenie sys0. Hodnotu parametra v_ngroups_allowed môžete zmeniť alebo získať z jadra alebo z databázy ODM. Hodnotu parametra z jadra používa systém, kým je spustený. Hodnota parametra z databázy ODM je platná po reštartovaní systému.

Získavanie povoleného počtu skupín z databázy ODM: Na získanie parametra v_ngroups_allowed musíte použiť príkazy alebo podrutiny. Na získanie parametra v_ngroups_allowed z databázy ODM musíte použiť príkaz **lsattr**.

Príkaz **lsattr** zobrazuje parameter v_ngroups_allowed ako atribút ngroups_allowed. V nasledujúcom príklade uvidíte ako sa má príkaz **lsattr** používať na získanie atribútu ngroups_allowed:

```
$ lsattr -El sys0
SW_dist_intr    false           Povoliť SW distribúciu prerušení                 True
autorestart     true            Automaticky REBOOTOVAŤ systém po zlyhaní         True
boottype        disk            nie je k dispozícii                              False
capacity_inc    1.00            Prírastok kapacity procesora                     False
capped          true            Oddiel je uzavretý                               False
conslogin       enable          Prihlásenie na systémovú konzolu                 False
cpuguard        enable          Sprievodca CPU                                   True
dedicated       true            Oddiel je vyhradený                              False
ent_capacity    4.00            Oprávnená kapacita procesora                     False
frequency       93750000        Frekvencia systémovej zbernice                   False
fullcore        false           Povoliť úplný výpis CORE                         True
fwversion       IBM,SPH01316    Verzia firmvéru a úrovne revízie                 False
iostat          false           Nepretržite udržiavať históriu I/O DISKU         True
keylock         normal          Stav zámku kľúča v čase zavedenia               False
max_capacity    4.00            Maximálna potenciálna kapacita procesora         False
max_logname     20              Maximálna dĺžka prihlásenia v čase zavedenia     True
maxbuf          20              Max. počet stránok v blok. CACHE I/O VYR. PAMÄTE  True
maxmbuf         0               Maximum kbajtov skutočnej pam. povolený pre MBUFS True
maxpout         0               Zn. VYSOKEJ hlad. pre čakajúci zápis I/O na 1 súbor  True
maxuproc        128             Maximálny počet PROCESOV povolených na užívateľa  True
```

```
min_capacity    1.00            Minimálna potenciálna kapacita procesora          False
minpout         0               Zn. NÍZKEJ hlad. pre čakajúci zápis I/O na 1 súbor  True
modelname       IBM,7044-270    Názov počítača                                    False
ncargs          6               Veľkosť zoznamu ARG/ENV v 4-kbajtových blockoch   True
pre430core      false           Použiť výpis JADRA štýlu pre-430                  True
pre520tune      disable         Režim kompatibility ladenia pre-520               True
realmem         3145728         Množstvo použiteľnej fyzickej pamäte v kB         False
rtasversion     1               Verzia RTAS otvoreného firmvéru                   False
sec_flags       0               Bezpečnostné príznaky                             True
sed_config      select          Režim SED (Stack Execution Disable)               True
systemid        IBM,0110B5F5F   Systémový identifikátor hardvéru                  False
variable_weight 0               Závažnosť kapacity premenného procesora           False
ngroups_allowed 128             Počet povolených skupín v čase zavedenia         True
$
```

Získavanie povoleného počtu skupín z jadra: Ak chcete z jadra získať parameter v_ngroups_allowed musíte použiť podrutinu sys_param.

```
#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() zlyhal rc = %d, číslo chyby = %d\n", rc, errno);
 }
```

Zmena povoleného počtu skupín v databáze ODM: Hodnotu pre Number of Groups Allowed musíte nakonfigurovať v jadre počas fázy zavedenia systému. Na zmenu hodnoty v databáze ODM použite príkaz **chdev**. Táto zmena sa prejaví po reštartovaní systému.

Ak chcete zmeniť parameter **v_ngroups_allowed** v databáze ODM pomocou príkazu **chdev**, napíšte:

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

Riadenie prístupu na základe rolí

Spravovanie systému je dôležitým aspektom každodennej prevádzky, a bezpečnosť je neoddeliteľnou súčasťou väčšiny funkcií spravujúcich systém. Okrem zabezpečenia operačného prostredia je tiež dôležité dôkladne sledovať každodenné aktivity systému.

Väčšina prostredí vyžaduje, aby rôzne povinnosti spojené so správou systému vykonávali odlišní užívatelia. Je dôležité zachovať oddelenie týchto úloh, aby ani jeden užívateľ nemohol náhodne ani úmyselne obísť zabezpečenie systému. Zatiaľ čo tradičnou administráciou systému UNIX nie je možné tieto ciele dosiahnuť, riadenie prístupu na základe rolí (RBAC) to umožňuje.

Tradičné obmedzenia pri správe systému UNIX

Niektoré tradičné problémy so správou systému UNIX rieši RBAC. K týmto problémom patria:

administračné konto root

Systém AIX a ostatné operačné systémy UNIX majú zvyčajne zadefinované jedno konto administrátora systému s názvom **root** (zvyčajne so stanoveným UID hodnoty 0), ktorý môže v systéme vykonávať všetky privilegované úlohy

správy systému. Spoliehanie sa na jedného užívateľa pri všetkých úlohách správy systému je problémom v súvislosti s delením povinností. Zatiaľ čo jedno administračné konto je v určitých prostrediach prijateľné, mnohé prostredia vyžadujú viac administrátorov, pričom každý administrátor je zodpovedný za iné úlohy správy systému.

Aby bolo možné zdieľať zodpovednosti týkajúce sa správy s viacerými užívateľmi systému, praxou v minulosti bolo buď zdieľať heslo užívateľa root alebo vytvoriť ďalšieho užívateľa s rovnakým UID, ako má užívateľ root. Táto metóda zdieľania povinností správy systému predstavuje bezpečnostné problémy, lebo každý administrátor má úplnú kontrolu nad systémom a neexistuje žiadna metóda na obmedzenie operácií, ktoré môže administrátor vykonávať. Keďže užívateľ root je najprivilegovanejší užívateľ, užívatelia root môžu vykonávať neautorizované operácie a môžu tiež vymazať všetky audity o týchto aktivitách, čím znemožnia sledovanie týchto administračných krokov.

Eskalácia privilégií cez SUID

Riadenie prístupu v operačných systémoch UNIX sa v minulosti vykonávalo pomocou UID priradeného k procesu na určovanie prístupu. UID užívateľa root hodnoty 0 však zvyčajne malo povolené obchádzať kontroly oprávnení. Preto proces spustený ako užívateľ root môže úspešne absolvovať všetky kontroly prístupu a vykonávať všetky operácie. To predstavuje bezpečnostný problém pre koncept UNIX aplikácií **setuid**.

Koncept **setuid** umožňuje spustenie príkazu pod inou identitou ako užívateľ, ktorý vyvolal príkaz. To je potrebné, keď bežný užívateľ potrebuje vykonať privilegovanú úlohu. Príkladom tejto situácie je príkaz AIX **passwd**. Keďže bežný užívateľ nemá prístup k súboru, ktorý obsahuje heslá užívateľov, je potrebné ďalšie privilégium na zmenu hesla užívateľa, takže príkaz **passwd** je **setuid** pre užívateľa root. Keď bežný užívateľ spustí príkaz **passwd**, operačnému systému sa bude zdať, ako keby k súboru pristupoval užívateľ root a tento prístup sa udelí.

Hoci tento koncept poskytuje požadované funkcie, prináša sprievodné riziko. Keďže program **setuid** účinne pracuje v kontexte užívateľa root, ak útočník úspešne prevezme program pred jeho ukončením, získa všetky právomoci užívateľa root a potom môže obchádzať všetky kontroly prístupu v operačnom systéme a vykonávať všetky operácie. Lepším riešením je priradiť programu len podmnožinu privilégií užívateľa root, aby sa dodržal “Princíp minimálnych privilégií” na strane 79 a hrozba sa zmiernila.

Elementy RBAC

RBAC umožňuje vytvoriť roly pre správu systému a delegovať administračné úlohy členom množiny dôveryhodných systémových užívateľov. V systéme AIX RBAC poskytuje mechanizmus, prostredníctvom ktorého môžu byť administračné úlohy, obvykle vyhradené len pre koreňového užívateľa, priradené normálnym užívateľom systému.

RBAC to dosiahne tak, že definuje pracovné funkcie (roly) v rámci organizácie a tieto roly potom priraďuje špecifickým užívateľom. RBAC v zásade vytvára rámec, ktorý umožňuje správu systému prostredníctvom rolí. Roly sú obvykle zadefinované s rozsahom umožňujúcim spravovať jeden alebo viac administračných aspektov prostredia. Priradenie roly užívateľovi mu účinne odovzdá sadu oprávnení alebo privilégií a práv. Napríklad jedna administračná rola sa môže vzťahovať na správu súborových systémov, iná rola môže povoľovať vytváranie užívateľských kont.

Správa cez RBAC má v porovnaní s klasickou správou systému UNIX nasledujúce výhody:

- Na správe systému sa môže podieľať viacero užívateľov bez toho, že by museli zdieľať prístup k jednému kontu.
- Odstupňovaná správa napomáha izolácii systému, pretože žiadnemu administrátorovi nie je potrebné poskytnúť vyššie oprávnenia, než je nevyhnutné.
- RBAC umožňuje presadzovanie bezpečnostného modelu s minimálnymi privilégiami. Užívatelia a aplikácie získajú potrebné privilégiá iba vtedy, keď ich potrebujú, čím sa redukuje možný dopad pri útoku na systém.
- Umožňuje zaviesť a vynútiť konzistentné bezpečnostné politiky v celej firme, pokiaľ ide o správu systému a riadenie prístupov.
- Definíciu roly stačí vytvoriť raz a potom ju podľa potreby priraďovať alebo odoberať užívateľom podľa toho, ako sa menia ich pracovné funkcie.

Prístup s RBAC sa zameriava na tri ústredné koncepty:

- Autorizácie

- Roly
- Privilégiá

Tieto koncepty spoločne umožňujú systému využívajúcemu RBAC presadzovať zásadu čo najnižších privilégií.

Autorizácie:

Autorizácia je textový reťazec asociovaný s funkciami alebo príkazmi, súvisiacimi s bezpečnosťou. Autorizácia predstavuje mechanizmus poskytovania práv užívateľom na vykonávanie privilegovaných akcií a na pridelenie rôznych úrovní funkčnosti rôznym triedam užívateľov.

Keď je spustený príkaz, na ktorý sa vzťahuje autorizácia, prístup bude poskytnutý len v prípade, keď má volajúci užívateľ požadovanú autorizáciu. Autorizáciu môžeme chápať ako kľúč, ktorý dokáže odomknúť prístup k jednému alebo viacerým príkazom. Autorizácie nie sú priamo priradené užívateľom. Užívateľom sú priradené roly, ktoré sú kolekciami autorizácií.

Roly:

Roly umožňujú vzájomné zoskupovanie množiny funkcií správy v systéme. Pri analógii, že autorizácia je kľúč, rola môže byť krúžok na kľúče, ktorý môže obsahovať viac autorizácií. Autorizácie je možné priamo priradiť roly alebo nepriamo priradiť pomocou subrole. Subrola je jednoducho ďalšia rola, z ktorej daná rola dedí autorizácie.

Samotná rola neudeľuje užívateľovi žiadne ďalšie právomoci, ale skôr slúži ako zhromažďovací mechanizmus pre autorizácie a zariadenie na priraďovanie autorizácií užívateľovi. Definovaním roly a priradením roly užívateľovi sa stanovia úlohy správy systému, ktoré môže užívateľ vykonávať. Po zadefinovaní roly môže administrátor roly priradiť túto rolu jednému alebo viacerým užívateľom na spravovanie privilegovaných operácií, ktoré táto rola reprezentuje. Okrem toho možno užívateľovi priradiť viacero rolí. Keď bola rola priradená užívateľovi, tento užívateľ môže pomocou autorizácií priradených roli odomknúť prístup k administračným príkazom v systéme.

Spôsob priradzovania rolí užívateľom určujú organizačné politiky a procedúry. Nepriradzujte príliš veľa autorizácií jednej roli ani nepriradzujte rolu príliš veľa užívateľom. Väčšina rolí by sa mala priraďovať len členom administračného personálu. Tak ako boli právomoci užívateľa root v minulosti udeľované dôveryhodným užívateľom, roly by sa mali priraďovať len dôveryhodným užívateľom. Udeľujte roly len užívateľom s legitímnymi potrebami a len na potrebnú dobu. Táto prax znižuje riziko, že autorizácie získa alebo zneužije neoprávnený užívateľ.

Privilégiá:

Privilégium je atribút procesu, ktorý umožňuje procesu prechádzať cez určité systémové zákazy a obmedzenia.

Mechanizmus privilégia poskytuje dôveryhodným aplikáciám schopnosti, ktoré nie sú povolené nedôveryhodným aplikáciám. Napríklad privilégiá možno používať na nahradenie bezpečnostných obmedzení, na povolenie rozšíreného používania určitých systémových prostriedkov, napríklad pamäte a diskového priestoru a na nastavenie výkonu a priority procesu. O privilégiu možno uvažovať ako o schopnosti, ktorá umožňuje procesu prekonať určité bezpečnostné obmedzenia v systéme.

Autorizácie a roly sú nástroje na úrovni užívateľa, ktoré konfigurujú schopnosť užívateľa pristupovať k privilegovaným operáciám. Na druhej strane sú privilégiá obmedzovacím mechanizmom v jadre na určovanie, či má proces povolené vykonávať konkrétnu operáciu.

Privilégiá sú prepojené s procesom a väčšinou sa získavajú vyvolaním privilegovaného príkazu. Vďaka týmto priradeným privilégiám je proces spôsobilý vykonávať príslušnú privilegovanú operáciu. Napríklad, ak užívateľ použije rolu, ktorá má autorizáciu na spustenie príkazu, pri spustení tohto príkazu sa procesu priradí množina privilégií.

Princíp minimálnych privilégií:

V operačnom systéme sú niektoré operácie privilegované, a ich vykonávanie sa obmedzuje na autorizovaných užívateľov. Tieto privilegované operácie obvykle zahŕňajú úlohy typu opätovné zavedenie systému, pridávanie a modifikovanie systémov súborov, pridávanie a odstraňovanie užívateľov a zmena systémového času a dátumu.

V tradičných systémoch UNIX môže byť proces alebo užívateľ v normálnom režime alebo v privilegovanom režime (čo sa tiež označuje ako superužívateľ alebo koreňový užívateľ). Proces spustený ako koreňový proces môže spúšťať akékoľvek príkazy a vykonávať akékoľvek systémové operácie, zatiaľ čo normálny užívateľ nemôže vykonávať privilegované operácie. Tradičný systém UNIX využíva veľmi hrubý koncept privilégií typu všetko alebo nič, a je vystavený bezpečnostnému riziku, ktoré predstavuje administrátor s príliš veľkými privilégiami.

Tradičný prístup v systéme UNIX, kde jeden privilegovaný režim poskytuje všetok prístup k systému, je príliš hrubý na to, aby splnil požiadavky na výrazne zabezpečený systém. Systém, ktorý má byť bezpečný, požaduje, aby bola každému procesu udelená čo najužšia sada privilégií, potrebných na vykonanie určitej úlohy. Privilégiá poskytujú tú výhodu, že ich stačí poskytnúť iba procesom, ktoré ich potrebujú. Takéto obmedzenie privilégií sa označuje ako zásada čo najmenšieho možného privilégia, a dokáže obmedziť potenciálne poškodenie systému, spôsobené nedbalým alebo zlomyseľným administrátorom či operátorom.

Napríklad zmena hesla vyžaduje určité privilégiá na prístup k súborom, ktoré obvykle nie sú prístupné bežným užívateľom. Keby mali užívatelia tieto privilégiá trvale, mohli by tiež vykonávať iné akcie, ktoré z bezpečnostného hľadiska nie sú vítané. Požadované privilégiá sa preto poskytujú iba príkazu **passwd**, a nie všetkým užívateľom.

V prostredí RBAC nemajú užívatelia sami osebe žiadne inherentné privilégiá. Užívatelia majú len povolenie spustiť určitý príkaz, a tomuto príkazu je potom poskytnuté privilégium. Ak by boli namiesto toho privilégiá poskytované priamo užívateľom, títo by ich mohli použiť v ľubovoľnom čase a ľubovoľným spôsobom. Obmedzenie privilégií na jednotlivé príkazy vytvára obmedzený kontext, v ktorom môžu byť privilégiá použité. To vedie k dokonalejšej bezpečnosti, pretože ak potenciálny útočník zneužije dôveryhodnú aplikáciu, bude mať len obmedzenú sadu privilégií namiesto plnej moci koreňového užívateľa so všetkými privilégiami.

Dôveryhodné aplikácie je potrebné dôkladne preskúmať predtým, ako im budú pridelené privilégiá. Navyše, privilégiá by mali byť pridelené len vtedy a tam, kde to aplikácia vyžaduje. Dôveryhodné aplikácie sú také isté ako ostatné programy, jediný rozdiel je, že môžu vykonávať akcie, ktoré sú nedôveryhodným aplikáciám odopreté.

RBAC v systéme AIX

Systém AIX poskytoval pred uvedením vydania AIX 6.1 iba obmedzenú implementáciu RBAC.

Počnúc verziou AIX 6.1 je k dispozícii nová implementácia RBAC, ktorá poskytuje veľmi jemne odstupňované mechanizmy na rozčlenenie úloh správy systému. Keďže tieto dve implementácie RBAC sa výrazne líšia čo do funkčnosti, zaužívali sa tieto dva termíny:

Klasický režim RBAC
: Historické správanie sa rolí v systéme AIX vzťahujúce sa na verzie staršie ako AIX 6.1

Rozšírený režim RBAC
: Nová implementácia predstavená vo verzii AIX 6.1

Podporované sú oba režimy fungovania. Predvolenou voľbou na novo inštalovanom systéme AIX 6.1 je rozšírený režim RBAC. V nasledujúcich témach sú stručne rozobraté oba tieto režimy a rozdiely medzi nimi spolu s informáciami o konfigurácii systému pre požadovaný režim RBAC.

Klasický režim RBAC:

Pred vydaním AIX 6.1 systém AIX poskytoval iba obmedzenú funkčnosť RBAC, ktorá umožňovala užívateľom iným ako root vykonávať niektoré úlohy správy systému.

V tejto implementácii RBAC, keď nekoreňový užívateľ vyvolal administračný príkaz, kód v príkaze určil, či bude užívateľovi priradená rola s požadovanou autorizáciou. Ak sa našla zhoda, príkaz sa ďalej vykonal. Ak sa nenašla, príkaz zlyhal s chybou. Často sa vyžaduje, aby príkaz, ktorý sa riadi autorizáciou, bol **setuid** pre koreňového užívateľa, aby autorizovaný vyvolávač mal potrebné privilégium na dokončenie operácie.

Táto implementácia RBAC tiež priniesla preddefinovanú sadu autorizácií, ktorú však užívatelia môžu rozšíriť, pričom tieto autorizácie určujú prístup k administračným príkazom. Okrem toho je poskytnutý aj rámec administračných príkazov a rozhraní na vytváranie rolí, priraďovanie autorizácií k rolám a priraďovanie rolí užívateľom.

Zatiaľ čo táto implementácia poskytuje schopnosť čiastočného delenia správcovských povinností, funguje s týmito obmedzeniami:

1. Tento rámec vyžaduje zmeny príkazov a aplikácií, aby boli povolené pre RBAC.
2. Preddefinované autorizácie nie sú odstupňované a mechanizmy na vytváranie autorizácií nie sú dostatočne robustné.
3. Na spustenie príkazu sa často požaduje členstvo v určitej skupine plus rola s príslušným oprávnením.
4. Oddelenie povinností sa ťažko prevádza do praxe. Ak má užívateľ priradených viacero rolí, nijako nemôže konať len pod jednou rolou. Takýto užívateľ má vždy všetky autorizácie pre všetky svoje roly.
5. Zásada čo najmenších možných privilégií sa v operačnom systéme neuplatňuje. Príkazy musia byť pre koreňového užívateľa SUID.

Klasický režim RBAC je z dôvodov kompatibility naďalej podporovaný, no štandardný režim RBAC je rozšírený režim. Na systéme AIX je preferovaný rozšírený režim RBAC.

Rozšírený režim RBAC:

Výkonnejšiu implementáciu RBAC poskytuje systém AIX 6.1. Aplikácie, ktoré na určité operácie vyžadujú administračné privilégiá, majú k dispozícii nové voľby integrácie s rozšírenou infraštruktúrou AIX RBAC.

Tieto integračné voľby sa zameriavajú na používanie jemne rozlíšených privilégií a autorizácií a na schopnosť nakonfigurovať akýkoľvek príkaz na systéme ako privilegovaný príkaz. Funkcie rozšíreného režimu RBAC sa nainštalujú a povolia štandardne na všetkých inštaláciách systému AIX počnúc verziou AIX 6.1.

Rozšírený režim RBAC ponúka konfigurovateľnú sadu autorizácií, rolí, privilegovaných príkazov, zariadení a súborov prostredníctvom nasledujúcich databáz RBAC. Pri rozšírenom RBAC môžu byť databázy trvalo umiestnené buď v lokálnom súborovom systéme, alebo sa dajú vzdialene riadiť prostredníctvom LDAP.

- Databáza autorizácií
- Databáza rolí
- Databáza privilegovaných príkazov
- Databáza privilegovaných zariadení
- Databáza privilegovaných súborov

Rozšírený režim RBAC prináša nové pomenúvacie konvencie pre autorizácie, ktoré umožňujú vytvorenie hierarchie autorizácií. AIX poskytuje sadu rozlíšených, systémom definovaných autorizácií, a administrátori môžu podľa potreby voľne vytvárať dodatočné užívateľom definované autorizácie.

Správanie rolí bolo zdokonalené, takže teraz umožňujú oddelenie funkčnosti v súvislosti s rolou. Rozšírené RBAC prináša koncept relácií rolí. Relácia roly je proces s jednou alebo viacerými asociovanými rolami. Užívateľ môže vytvoriť reláciu roly pre ľubovoľné roly, ktoré mu boli priradené, takže môže naraz aktivovať jednu rolu alebo niekoľko vybratých rolí. Štandardne platí, že nový systémový proces nemá priradené žiadne roly. Roly boli ďalej zdokonalené tak, aby podporovali požiadavku, že užívateľ sa musí autentifikovať skôr, ako aktivuje rolu. Chráni to pred nebezpečenstvom, že útočník preberie reláciu užívateľa, pretože takto sa bude musieť autentifikovať, aby mohol aktivovať rolu užívateľa.

Nový prvok - databáza privilegovaných príkazov - uplatňuje zásadu čo najmenších možných privilégií. Odstupňovanie systémových privilégií je jemnejšie, a explicitné privilégiá môžu byť poskytnuté príkazu a vykonanie príkazu môže riadiť autorizácia. Tieto funkcie umožňujú vynucovať kontroly autorizácie pri spúšťaní príkazov bez toho, aby bolo potrebné meniť samotný kód príkazu. Použitie databázy privilegovaných príkazov eliminuje potrebu aplikácií SUID a SGID, pretože je možné priraďovať iba vyžadované privilégiá.

Databáza privilegovaných zariadení povoľuje prístup k zariadeniam, ktoré majú byť riadené privilégiami, a databáza privilegovaných súborov umožňuje užívateľom bez potrebných privilégií pristupovať k obmedzeným súborom na základe autorizácie. Tieto databázy zvyšujú granularitu úloh správy systému, ktoré tak môžu byť priradené aj užívateľom, ktorým inak chýbajú potrebné privilégiá.

Informácie v databázach RBAC sa zhromažďujú a overujú a následne sa odošlú do oblasti kernelu, ktorá sa označuje ako Kernel Security Tables (KST). Je dôležité poznamenať, že stav údajov v bezpečnostných tabuľkách KST určuje bezpečnostnú politiku pre systém. Položky, ktoré sa menia na užívateľskej úrovni databáz RBAC, sa nepoužívajú pri rozhodovaní o zabezpečení, pokiaľ tieto údaje neboli odoslané do tabuliek KST príkazom **setkst**.

Konfigurácia režimu RBAC:

Režim RBAC je riadený konfiguračnou premennou v kerneli, platnou pre celý systém. Premenná určuje, či je povolený alebo zakázaný rozšírený režim RBAC.

Rozšírený režim RBAC je v systémoch AIX 6.1 a novších štandardne povolený. Môžete spustiť príkaz **chdev** na zariadení **sys0** a zadať hodnotu **false** pre atribút **enhanced_RBAC**, čím zakážete rozšírený režim RBAC a vrátite sa ku klasickému režimu RBAC. Aby bola zmena atribútu **enhanced_RBAC** účinná, musíte opätovne zaviesť systém. Ak chcete povoliť rozšírený režim RBAC, nastavte atribút **enhanced_RBAC** na hodnotu **true**. Z hľadiska programovania sa tento režim dá nastaviť alebo dotazovať aj prostredníctvom systémového volania **sys_parm()**.

Na systéme spustite nasledujúci príkaz a získajte aktuálny režim RBAC:

```
lsattr -E -l sys0 -a enhanced_RBAC
```

Rozšírený režim RBAC môžete zakázať spustením nasledujúceho príkazu a následným opätovným zavedením systému:

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

V prostredí WPAR sa dá režim RBAC konfigurovať iba z globálneho systému, a jednotne ovplyvní globálny systém aj všetky WPAR na systéme.

Porovnanie klasického a rozšíreného režimu RBAC:

Existujúce aj nové rozhrania boli zmenené tak, aby si overili konfiguráciu systému a potom spustili nový kód, alebo fungovali starým spôsobom.

V klasickom režime RBAC sú vynucované iba autorizácie, ktoré sa kontrolujú priamo v rámci kódu samotného príkazu. Bezpečnostné tabuľky kernelu (KST) nemajú žiaden vplyv na vykonávanie príkazov ani kontrolu autorizácie. Rozhodnutie, či má užívateľ autorizáciu, sa drží správania klasického režimu RBAC, kde sa získavajú všetky autorizácie užívateľa a hľadá sa zhoda. Nové funkcie, ako je príkaz **swrole** a atribúty **default_roles** a **auth_mode**, nie sú v klasickom režime RBAC k dispozícii. Klasický režim RBAC však napriek tomu podporuje nové privilégiá, autorizácie a riadenie príkazov.

V nasledujúcej tabuľke môžete vidieť niektoré rozdiely medzi klasickým a rozšíreným režimom RBAC.

Tabuľka 9. rozdiely medzi klasickým a rozšíreným režimom RBAC

Vlastnosť	Klasické RBAC	Rozšírené RBAC
Aktivovanie rolí	Všetky roly užívateľa sú stále aktívne	Roly štandardne nie sú aktívne, kým nie sú explicitne nadobudnuté prostredníctvom príkazu **swrole**
Atribút **default_roles**	Nie je k dispozícii	Podporovaný
Príkaz **swrole**	Nie je k dispozícii	Podporovaný
Príkazy správy rolí	Podporovaný	Podporovaný
Príkazy správy autorizácií	Podporovaný	Podporovaný
Hierarchia autorizácií	Každá autorizácia je nezávislá. Hierarchia neexistuje.	Podporuje koncept hierarchie autorizácií, kde jedna autorizácia môže byť rodičom pre iné autorizácie
Kontrola autorizácie	Uplatňovaná, iba keď samotný príkaz kontroluje autorizáciu	Uplatňovaná prostredníctvom databázy privilegovaných príkazov a/alebo samotného príkazu
Odstupňované privilégiá	Podporovaný	Podporovaný
Príkaz **pvi**	Nie je k dispozícii	Podporovaný
Bezpečnostné tabuľky kernelu (KST)	Nie je k dispozícii	Podporovaný
Umiestnenie databáz RBAC	Lokálne súbory	Lokálne súbory alebo LDAP

Používanie rozšíreného RBAC

Aby mohli administrátori systému efektívne využívať rozšírený RBAC, mali by mať znalosti v nasledujúcich oblastiach.

Autorizácie RBAC:

Autorizácie sú dôležitou súčasťou Riadenia prístupu podľa rolí (Role Based Access Control, RBAC). Operačný systém používa reťazce autorizácií na určovanie spôsobilosti pred vykonaním privilegovaných operácií. Súvisiace kontroly je možné vykonávať explicitne v kóde alebo pomocou zavádzača pri spúšťaní privilegovaných spustiteľných programov.

Pomenúvanie autorizačných reťazcov označuje privilegovanú operáciu, ktorú reprezentujú a riadia. Pomenúvacie konvencie systému AIX pre autorizácie podporujú hierarchickú štruktúru, ktorá je určená textovým názvom autorizácie. Autorizačné reťazce systému AIX používajú na popis hierarchie autorizácií formát zápisu s bodkami. Napríklad, autorizácia na vytváranie nových súborových systémov je **aix.fs.manage.create**. Ak je táto autorizácia súčasťou určitej roly, potom užívateľ, ktorý má priradenú túto rolu, môže vytvárať súborové systémy AIX. Ak je rodičovská autorizácia **aix.fs.manage** súčasťou určitej roly, potom užívateľ, ktorý má priradenú túto rolu, môže vykonávať ostatné úlohy správy súborového systému, ako aj vytvárať súborové systémy.

AIX RBAC rozlišuje medzi systémom zabezpečenými autorizáciami (systémom definovanými autorizáciami) a autorizáciami vytvorenými po inštalácii (užívateľom definovanými autorizáciami).

Systémom definované autorizácie:

AIX poskytuje množinu preddefinovaných a nemodifikovateľných autorizácií. Tieto sú známe ako systémom definované autorizácie. Tieto autorizácie sú priradené k rôznym privilegovaným operáciám systému AIX. Toto priradenie je špecifikované v Databáze privilegovaných príkazov.

Na vrchole hierarchie systémom definovaných autorizácií je autorizácia **aix**. Táto autorizácia je rodičom všetkých ostatných systémom definovaných autorizácií. Udelením tejto autorizácie nejakej roli sa tejto roli udelí každá systémom definovaná autorizácia. Ak chcete zobraziť kompletnú množinu systémom definovaných autorizácií systému AIX a stručný popis každej autorizácie, spustite nasledujúci príkaz:

```
lsauth –f –a description ALL_SYS
```

Výstup vyššie uvedeného príkazu informuje o tom, že zoznam systémom definovaných autorizácií má viacúrovňovú hierarchiu. Napríklad, autorizácia **aix** má niekoľko priamych potomkov. Každý z týchto potomkov je potom rodičom inej hierarchie. Autorizácia **aix.fs** obsahuje niekoľko dcérskych autorizácií, vrátane **aix.fs.manage**, ktorá zase obsahuje niekoľko autorizácií ako **aix.fs.manage.change** a **aix.fs.manage.create**.

Užívateľom definované autorizácie:

Okrem systémom definovaných autorizácií umožňuje RBAC systému AIX administrátorom systému definovať vlastné voliteľné autorizácie v databáze autorizácií (/etc/security/authorizations). Tie sú známe ako užívateľom definované autorizácie.

Administrátor systému môže pridávať, upravovať alebo vymazávať užívateľom definované autorizácie. Napríklad, administrátor systému môže povoliť niektorým užívateľom spustiť privilegovaný príkaz vytvorením užívateľom definovanej autorizácie a následným priradením tejto autorizácie k príkazu a udelením tejto autorizácie roli, ktorá je priradená týmto užívateľom.

Užívateľom definované autorizácie podporujú rovnaký koncept hierarchie ako systémom definované autorizácie. Sú však isté obmedzenia týkajúce sa pomenúvania užívateľom definovaných autorizácií systému AIX.

- Užívateľom definované autorizácie musia byť definované pod novým rodičom najvyššej úrovne. Inými slovami, užívateľom definované autorizácie nesmú byť potomkom systémom definovaných autorizácií (**aix**).
- Názov autorizácie môže obsahovať maximálne 63 tlačiteľných znakov.
- Hierarchia rodiča autorizácie môže obsahovať maximálne osem úrovní.
- Autorizácia môže mať ľubovoľný počet priamych potomkov, ale môže mať len jedného priameho rodiča. Nezávislé autorizácie nemôžu mať rovnakého priameho potomka.

Keďže táto hierarchia nepovoľuje mať viacero priamych rodičov, nemôžete vytvoriť užívateľom definovanú autorizáciu, ktorá je rodičom existujúcej systémom definovanej autorizácie. preto pokus o vytvorenie autorizácie s názvom **aix.custom** zlyhá a vytvorenie autorizácie s názvom **custom.aix** vytvorí úplne novú autorizáciu a nefunguje ako rodič systémom definovanej autorizácie **aix**.

Pri vytváraní užívateľom definovanej autorizácie sa navrhuje nasledujúca syntax, aby nedochádzalo ku konfliktom medzi názvami autorizácií medzi viacerými softvérovými komponentmi:

*názov_dodávateľa***.***názov_produktu***.***funkcia***.***funkcia1***.***funkcia2*...

názov_dodávateľa
: Identifikuje názov dodávateľa softvérového modulu.

názov_produktu
: Názov produktu vyššej úrovne pre produkt riadený pomocou RBAC.

*funkcia***,** *funkcia1***,** *funkcia2* **...**
: Tieto reťazce reprezentujú funkcie, ktoré sú riadené pomocou RBAC. Tieto reťazce poskytujú tiež hierarchickú reprezentáciu spôsobu organizácie týchto funkcií.

Napríklad, **ibm.db2.manage** by mohol potenciálne reprezentovať správne aspekty databázového balíka IBM DB2. Ako už bolo spomenuté, reťazec pre *názvov_dodávateľa* **aix** je vyhradený pre použitie systémom AIX a nie je povolený pre užívateľom definované autorizácie.

Existuje niekoľko príkazov na správu autorizácií, pomocou ktorých môžu administrátori systému vypisovať, vytvárať, upravovať a odstraňovať užívateľom definované autorizácie. Užívateľom definované autorizácie možno vytvárať pomocou príkazu **mkauth**, modifikovať pomocou príkazu **chauth** odstraňovať pomocou príkazu **rmauth** a zobrazovať pomocou príkazu **lsauth**. Ak chcete zobraziť všetky užívateľom definované autorizácie a stručný popis každej autorizácie, spustite nasledujúci príkaz:

```
lsauth –f –a description ALL_USR
```

Pred vytvorením užívateľom definovanej autorizácie vezmite do úvahy nasledujúce položky:

- Bolo by namiesto vytvorenia novej užívateľom definovanej autorizácie vhodné použiť existujúcu užívateľom definovanú autorizáciu?
- Patrí nová autorizácia pod existujúcu hierarchiu užívateľom definovaných autorizácií alebo je to prvá autorizácia novej hierarchie?
- Ak je to nová hierarchia, aká je jej štruktúra?
- Aký je textový popis autorizácie?
- Vyžaduje sa jazykový preklad popisu autorizácie?
- Je nejaký dôvod pri vytváraní tejto autorizácie špecifikovať určité ID autorizácie? Odporúča sa na vygenerovanie ID autorizácie použiť príkaz **mkauth**.

Po zohľadnení týchto položiek vykonajte nasledujúce kroky na vytvorenie autorizácie:

1. Ak sa vyžaduje jazykový preklad, vytvorte alebo pridajte popis do katalógu správ.
2. Pomocou príkazu **mkauth** vytvorte všetky rodičovské autorizácie v hierarchii, ak ešte neexistujú.
3. Pomocou príkazu **mkauth** vytvorte požadovanú autorizáciu. Ak sa vyžaduje konkrétna hodnota, s príkazom zadajte atribút **id**.

Migrácia starších, klasických autorizácií:

Pred verziou AIX verzie 6.1 mal operačný systém obmedzenú, preddefinovanú sadu autorizácií, ktoré operačný systém rozpoznával. Tieto autorizácie neboli zadefinované v žiadnom súbore v systéme, dali sa však pohotovo priraďovať rolám. Na podporu týchto tradičných autorizácií v novom rámci RBAC operačného systému AIX verzie 6.1 a novších rámcoch RBAC sú tieto tradičné autorizácie definované ako užívateľom definované autorizácie a štandardne sa poskytujú v autorizačnej databáze.

Keďže operačný systém AIX prechádza na nové pomenúvacie konvencie autorizácie, všetky kontroly starých názvov autorizácie v operačnom systéme AIX boli zmenené tak, aby kontrolovali aj nové súvisiace autorizácie a povolili prístup, ak existuje hociktorá autorizácia pre proces. Nasledujúca tabuľka obsahuje zoznam starších (klasických) preddefinovaných autorizácií a im zodpovedajúcich nových, systémom definovaných autorizácií.

Autorizácia existujúca v systéme AIX	**Príslušná nová autorizácia**
Zálohovanie	aix.fs.manage.backup
Diagnostika	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

Roly RBAC:

Roly sú mechanizmus používaný na priraďovanie autorizácií užívateľovi a na vzájomné zoskupovanie množiny systémových administračných úloh. Rola systému AIX je primárne kontajner na zhromažďovanie autorizácií.

AIX podporuje priame priradenie autorizácií roli alebo nepriame priradenie autorizácií cez subrolu. Podrola sa dá špecifikovať pre rolu v atribúte **rolelist** roly. Nakonfigurovanie roly tak, aby mala určenú subrolu, efektívne priradí tejto roly všetky autorizácie v subrole.

Priradením roly užívateľovi sa umožní tomuto užívateľovi prístup k roli a používanie autorizácií obsiahnutých v tejto roli. Administrátor systému môže priradiť rolu viacerým užívateľom a priradiť viacero rolí jednému užívateľovi. Užívateľ, ktorému bolo priradených viac rolí, môže súčasne aktivovať viac ako jednu rolu (až po maximálne osem rolí), ak je to potrebné na vykonávanie funkcií správy systému.

AIX poskytuje množinu preddefinovaných rolí pre správu systému. Predpokladá sa však, že zákazníci si budú musieť vytvoriť vlastné voliteľné roly alebo upraviť existujúce preddefinované roly. K dispozícii je niekoľko súborov na správu rolí na výpisy, vytváranie, modifikáciu odstraňovanie rolí AIX. Roly možno vytvárať pomocou príkazu **mkrole**, modifikovať pomocou príkazu **chrole** odstraňovať pomocou príkazu **rmrole** a zobrazovať pomocou príkazu **lsrole**.

Pri vytváraní novej roly AIX vezmite do úvahy nasledujúce položky:

- Aký bude názov roly?
- Názov roly je textový reťazec, ale mal by poskytovať určitý náhľad do schopností roly. Názvy rolí môžu obsahovať maximálne 63 tlačiteľných znakov.
- Aké autorizácie sa vyžadujú pre rolu? Rozhodnite, či autorizácie majú byť priamo priradené roly alebo nepriamo priradené prostredníctvom subrole.
- Mal by sa užívateľ požiadať o autentifikáciu pri aktivovaní roly?

Aktivovanie roly:

V operačnom systéme AIX verzie 6.1 a neskorších vydaniach s rozšíreným zabezpečením RBAC k relácii užívateľa štandardne nie sú priradené žiadne roly alebo oprávnenia, keď sa užívateľ autentifikuje v systéme. Aby sa k relácii priradili roly, užívateľ musí vyvolať samostatný autentifikačný príkaz (**swrole**), aby sa prepla rola alebo roly.

Užívateľ môže aktivovať iba roly, ktoré mu boli predtým priradené. Štandardne keď užívateľ vstupuje do relácie roly alebo keď pridáva k svojej relácii rolu, musí sa autentifikovať ako on sám. Voliteľne je možné určiť roly, ktoré nevyžadujú autentifikáciu s atribútom roly **auth_mode**.

Prepnutie na novú reláciu roly vytvorí nový shell (reláciu) bez zdedenia rolí z predchádzajúcej relácie. Dosiahne sa to tak, že sa pre rolu vytvorí nový shell procesu a procesu sa priradí nové ID roly (RID). Vytvorenie novej roly je podobné použitiu príkazu **su**, až na to, že tu sa zmení iba ID roly procesu, a nie charakteristiky typu UID alebo GID. Príkaz **swrole** dovoľuje užívateľovi vytvoriť reláciu roly zloženú z jednej alebo viacerých rolí. Neexistuje žiadne obmedzenie, ktoré by užívateľovi bránilo prepnúť sa do novej relácie roly z aktuálnej relácie roly. Keďže nová relácia je nový proces, táto nová relácia nezdedí žiadne roly z predchádzajúcej relácie. Aby sa obnovila predchádzajúce relácia, užívateľ musí ukončiť aktuálnu reláciu roly. Roly prijaté v relácii (sada aktívnych rolí) sa dajú vypísať spustením príkazu **rolelist** v relácii. Administrátor môže použiť aj príkaz **rolelist** na vypísanie sady aktívnych rolí pre daný proces systému.

Užívateľovi môže byť voliteľne priradená predvolená sada rolí s novým užívateľským atribútom **default_roles**. Tento atribút je určený pre situácie, keď procesy, ktoré boli vytvorené pre užívateľa, musia byť vždy asociované s danou sadou rolí, napríklad pri príkaze **cron**. Príkaz cron beží na pozadí a spúšťa príkazy ako definovaný užívateľ. Je možné, že niektoré spúšťané príkazy vyžadujú autorizáciu. To vyžaduje schopnosť určiť, že istá sada rolí bude pre niektoré ID užívateľa vždy aktívna, pretože pre príkaz **cron** neexistuje mechanizmus pre neskoršie získanie týchto rolí. Atribút **default_roles** môže byť nastavený tak, aby obsahoval až osem názvov rolí alebo špeciálnu hodnotu **ALL**. Nastavenie **default_roles=ALL** priradí relácii všetky užívateľské roly. Ak bolo užívateľovi priradených viac ako osem rolí, pre reláciu bude povolených iba prvých osem rolí.

Maximálny počet rolí na reláciu:

V rozšírenom režime RBAC môže administrátor systému pre celý systém nakonfigurovať maximálny počet rolí, ktoré môže užívateľ aktivovať v určitej relácii roly. Štandardne môže užívateľ v relácii aktivovať maximálne osem rolí.

Niektoré prostredia môžu vyžadovať striktnejšie oddelenie povinností, takže užívateľ môže v jednej relácii aktivovať iba jednu rolu. V týchto prostrediach môžete zmeniť atribút **maxroles** v odseku **usw** v súbore /etc/security/login.cfg a obmedziť tak maximálny povolený počet rolí v jednej relácii. Atribút **maxroles** môžete nastaviť na hodnotu z rozsahu 1 až 8 a určiť tak maximálny povolený počet rolí na jednu reláciu.

Ak chcete zobraziť aktuálnu hodnotu obmedzenia počtu rolí v relácii, spustite nasledujúci príkaz:

```
lssec –f /etc/security/login.cfg –s usw –a maxroles
```

Ak chcete modifikovať systém, aby mohol užívateľ naraz aktivovať iba jednu rolu, spustite príkaz:

```
chsec –f /etc/security/login.cfg –s usw –a maxroles=1
```

Zmena hodnoty atribútu **maxroles** je okamžite účinná pre akékoľvek novo vytvorené relácie roly a nevyžaduje opätovné zavedenie systému. Relácie rolí, ktoré existovali ešte pred zmenou tejto hodnoty, táto zmena neovplyvní. Vynútenie maximálneho počtu rolí na reláciu sa vykoná pri spustení relácie.

Preddefinované roly:

V lokálnej databáze rolí (/etc/security/roles) v inštalácii operačného systému AIX verzie 6.1 a novších inštaláciách je definovaná preddefinovaná množina rolí. Táto sada rolí má za úlohu zoskupiť typické administračné povinnosti.

Táto sada rolí slúži ako pomôcka pre rozdelenie administračných úloh a povinností. Administrátori rolí môžu roly meniť alebo odstraňovať, prípadne vytvárať nové roly podľa potrieb v ich prostredí. Nasleduje zoznam poskytnutých rolí a stručný popis ich jednotlivých daností.

Názov roly	**Popis roly**
auditadm	Administrátor auditov. Rola auditadm je zodpovedná za konfiguráciu systémových politík auditovania a protokolovania, vrátane atribútov pre celý systém, pre jediného užívateľa a pre jedinú rolu. Táto rola má prístup k zobrazeniu záznamov auditu.
fsadm	Administrátor súborového systému. Rola fsadm vytvára súborové systémy a sprístupňuje ich užívateľom v systéme. Niektoré zo zodpovedností roly fsadm zahŕňajú: • Určovanie politík pripájania • Politiky zdieľania • Priraďovanie kvóty • Určovanie úrovne kompresie • Vytváranie formátov súborového systému • Vykonávanie aktivít zálohy a obnovy
isso	Bezpečnostný pracovník informačného systému. ISSO zodpovedá za vytváranie a priraďovanie rolí, ide preto o najmocnejšiu rolu v systéme. Niektoré zodpovednosti ISSO: • Vytvorenie a udržiavanie bezpečnostnej politiky • Nastavenie hesiel pre užívateľov • Sieťová konfigurácia • Správa zariadení
pkgadm	Administrátor softvérových balíkov. Rola pkgadm je zodpovedná za softvér, ktorý je nainštalovaný v systéme, a má predvolené oprávnenia na inštaláciu, aktualizáciu a odstránenie systémového softvéru.

Názov roly	Popis roly
sa	Administrátor systému. Rola SA poskytuje funkcie potrebné pre každodennú správu systému. SA zodpovedá za: • Správu užívateľov (okrem nastavovania hesiel) • Spravovanie súborových systémov • Aktualizácie nainštalovaného softvéru • Správu sieťových démonov • Vyhradenie zariadení
secadm	Bezpečnostný administrátor. Rola secadm udržiava nastavenia bezpečnosti v systéme. Rola secadm priraďuje užívateľom atribúty ako členstvo v skupinách, roly, autorizácie a odstránenia a priraďuje roly, ktoré ešte nie sú definované v ich rolách. Rola secadm tiež priraďuje bezpečnostné atribúty systémovým objektom, vrátane nastavení RBAC, zoznamov riadenia prístupu, vlastníctva a členstva. Niektoré zo zodpovedností roly secadm zahŕňajú nasledovné: • Priraďovanie hesiel novým užívateľským kontám • Odomykanie uzamknutých kont
so	Operátor systému. Rola SO poskytuje funkcie potrebné pre každodennú prevádzku systému. SA zodpovedá za: • Vypnutie a opätovné zavedenie systému • Zálohovanie a obnovu súborového systému a stanovenie kvót • Protokolovanie a sledovanie chýb systému a štatistiku • Riadenie pracovného zaťaženia
svcadm	Administrátor služieb. Rola svcadm umožňuje, konfiguruje a zakazuje systémové služby. Táto rola umožňuje konfiguráciu sieťových atribútov ako IP adresy, trasy, názvy hostiteľov a politiky brány firewall.
sysop	Operátor systému. Rola sysop udržiava celkový systém s oprávneniami, ktoré zahŕňajú spúšťanie systémovej diagnostiky a vykonávanie rutinnej systémovej údržby. Niektoré z úloh, za ktoré je rola sysop zodpovedná, zahŕňajú: • Úplné odstraňovanie protokolových súborov a tlačových frontov • Zastavenie a reštart systémov
useradm	Užívateľský administrátor. Rola useradm je zodpovedná za úlohy vyššej úrovne súvisiace s údržbou užívateľov bez správy hesiel. Rola useradm vytvára, upravuje a vymazáva kontá užívateľov tak, ako sú definované predvolenými bezpečnostnými nastaveniami. Táto rola tiež vytvára ďalšie roly a skupiny s predvolenými bezpečnostnými nastaveniami.

Migrácia rolí:

Ak sa systém AIX pred systémom AIX verzie 6.1 aktualizuje na úroveň rozšíreného RBAC systému AIX pomocou inštalácie migrácie, migrácia súboru /etc/security/roles sa pokúsi aktualizovať súbor na nové funkcie, pričom ponechá súčasné schopnosti rolí.

Definície rolí v tomto súbore sa zachovajú a jednoducho sa upravia tak, aby obsahovali jedinečné ID roly, aby táto rola fungovala správne v novom pracovnom rámci. Všetky autorizácie v súbore /etc/security/roles, ktoré nie sú známe preddefinované autorizácie, sa budú považovať za užívateľom definované autorizácie. Počas migrácie sa názvy týchto

autorizácií pridajú ako položky v lokálnej databáze autorizácií /etc/security/authorizations. Okrem migrácie starých definícií rolí sa do súboru pridajú nové preddefinované roly. Po migrácii musí administrátor systému overiť, či sú tieto autorizácie a roly definované ako potrebné pre prostredie.

Privilégiá RBAC:

Rozšírený rámec RBAC sa maximálne spolieha na systémové privilégiá povoľujúce neprivilegovaným užívateľom vykonávať privilegované úlohy. Privilégium je mechanizmus používaný na udeľovanie rozšírenej funkcionality pre proces v systémových volaniach.

Koncept privilégií je primárne konštrukcia na úrovni jadra, lebo definícia a väčšina kontrol sa vykonáva v jadre. Sú však k dispozícii rozhrania na úrovni užívateľov na spracovanie priradenia privilégií príkazom, zariadeniam a procesom.

Je dôležité poukázať na rozdiel medzi privilégiami a autorizáciami. Privilégiá aj autorizácie sa používajú riadenie určitých prípustných výnimiek v systémovej bezpečnostnej politike. Definujúcim rozdielom medzi privilégiami a autorizáciami je, že privilégiá sú priradené konkrétnym procesom, zatiaľ čo autorizácie sú priradené užívateľom cez roly. Autorizácie sú umiestnené pri roli a užívateľovi, ktorý má rolu a nezávisí od spúšťaného programu. Privilégiá sú umiestnené pri programe a zabezpečujú mechanizmus na jemné doladenie systémovej bezpečnostnej politiky. Vďaka týmto priradeným privilégiám je proces spôsobilý vykonávať príslušnú privilegovanú operáciu.

Privilégiá sú definované v jadre systému AIX ako jednotlivé bity bitovej masky, ktoré vynucujú riadenie prístupu nad privilegovanými operáciami. V systéme AIX je k dispozícii viac ako 100 privilégií, čo zabezpečuje veľmi jemnú granulovanú kontrolu nad privilegovanými operáciami. Pri zisťovaní prístupu v systémovom volaní jadru zistí, či proces má požadovaný bit príslušného privilégia a potom udelí alebo zamietne požiadavku.

Privilégiá sa priraďujú vyvolaniam príkazu prostredníctvom databázy privilegovaných príkazov a privilégiá sa používajú na riadenie prístupu k zariadeniam prostredníctvom databázy privilegovaných zariadení.

Pomenúvanie a hierarchia privilégií:

Privilégiá AIX nemôže vytvárať, modifikovať alebo vymazávať administrátor systému.

Zoznam dostupných privilégií a stručný popis privilégia možno zobraziť v systéme spustením nasledujúceho príkazu:

```
lspriv -v
```

Privilégiá poskytované v AIX sú vypísané v AIX privilégiách. Všetky privilégiá systému AIX majú textovú reprezentáciu bitu privilégia začínajúcu na **PV_**. Pomenúvacia konvencia použitá po predpone **PV_** znamená hierarchický vzťah medzi privilégiami. Napríklad, privilégium auditovania **PV_AU_** je rodičom privilégií **PV_AU_ADD**, **PV_AU_ADMIN**, **PV_AU_READ**, **PV_AU_WRITE** a **PV_AU_PROC**. Pri kontrole privilégia systém najskôr zistí, či proces má najnižšie potrebné privilégium a potom postupuje nahor po hierarchii a kontroluje prítomnosť silnejšieho privilégia. Privilégium **PV_ROOT** je špeciálne privilégium, ktoré reprezentuje rodiča všetkých privilégií okrem **PV_SU_**. Proces, ktorý je priradený privilégiu **PV_ROOT** sa správa tak, ako keby bol priradený každému privilégiu v systéme okrem **PV_SU_**.

Množiny privilégií procesu:

V jadre sa definuje viac množín privilégií na zabezpečenie rôznych ovládacích prvkov pre privilegované operácie. Viaceré množiny privilégií umožňujú operačnému systému vynútiť si dynamické ovládacie prvky privilégií a umožňujú aplikáciám riadiť princípy najmenšieho.

Privilégiá sú prepojené s procesmi prostredníctvom nasledujúcich množín privilégií:

Limiting Privilege Set (LPS)
: Definuje pevný limit na privilégiách pre daný proces. Žiadna eskalácia privilégií v systéme nemôže zvýšiť privilégiá procesu za túto hodnotu. To znamená, že proces nemôže získať žiadne ďalšie privilégiá, ako uvádza táto hodnota, pomocou žiadneho z definovaných systémových rozhraní. Inými slovami, tento proces je v

každom momente obmedzený na tieto privilégiá. To tiež znamená, že zvyšné množiny privilégií vždy budú podmnožiny LPS. Aj keď LPS nie je možné rozvinúť, každý proces bude mať právo zmenšiť LPS. Akonáhle sa však LPS zmenší, nie je možné ho znova rozvinúť na pôvodnú hodnotu. Zníženie LPS umožňuje procesu obmedziť hranice týkajúce sa priradených privilégií. Napríklad, proces by mohol zmenšiť LPS tesne pred spustením voliteľného užívateľom zabezpečeného programu. Štandardne sú všetky privilégiá dostupné v systéme množinou v LPS pre proces.

Maximum Privilege Set (MPS)
Úplná množina privilégií, ktoré je proces autorizovaný používať. MPS môže obsahovať ľubovoľné privilégium v LPS, ale nemôže prekročiť LPS. MPS sa môže počas životnosti procesu z mnohých príčin zmeniť. Nasledujú niektoré z týchto príčin:

- Keď aktuálny proces vykoná iný privilegovaný príkaz a potom získa príslušné dodatočné privilégiá
- Ak má proces to správne privilégium, potom môže rozvinúť MPS programovo dynamickým spôsobom

Effective Privilege Set (EPS)
Zoznam privilégií, ktoré sú momentálne aktívne pre proces. EPS je vždy podmnožinou MPS procesu a používa ho jadro na vykonávanie kontrol prístupu týkajúcich sa privilegovaných operácií. S EPS je možné manipulovať podľa procesu a môže sa rovnať MPS, ale nemôže prekročiť MPS. Dynamickú manipuláciu s EPS je možné vykonávať podľa procesu na posilnenie princípov privilégia najmenšieho. Napríklad, kód užívateľského priestoru môže potenciálne zvýšiť bit privilégia auditu v EPS pomocou API **priv_raise** pred vykonaním volania systému týkajúceho sa auditu alebo volania jadra. Toto privilégium je možné potom znížiť pomocou API **priv_lower**, keď sa volanie auditu vráti.

Inheritable Privilege Set (IPS)
Privilégiá, ktoré sa odovzdávajú z rodičovského procesu na MPS a EPS jeho dcérskych potomkov. IPS môže obsahovať ľubovoľné privilégium v LPS, ale nemôže prekročiť LPS. IPS je možné nastaviť v procese nasledujúcimi spôsobmi:

- Ak má proces to správne privilégium, potom môže rozvinúť IPS programovo pomocou systémového volania **setppriv**
- Keď sa spustí privilegovaný príkaz, privilégiá špecifikované v atribúte **inheritprivs**, ktorý je priradený príkazu, sa priradia IPS.

Used Privilege Set (UPS)
Znamená privilégiá, ktoré boli použité pre kontroly prístupu počas životnosti procesu. UPS je možné používať na určovanie privilégií vyžadovaných procesom. Keď jadro skontroluje, či proces má dané privilégium, uloží úspešnú kontrolu do UPS pre toto privilégium.

Workload Partition Privilege Set (WPS)
Systémový WPAR je možné obmedziť tak, aby nepovoľoval privilegované operácie, ktoré sú povolené v globálnom WPAR. Privilegované operácie povolené v systémovom WPAR je možné riadiť prostredníctvom WPS. Globálny užívateľ môže priradiť obmedzenú množinu privilégií pre WPAR pomocou WPS. WPS je možné špecifikovať v konfiguračnom súbore /etc/wpar/secattrs alebo počas spúšťania WPAR pomocou príkazu **/usr/sbin/startwpar**. Všetky procesy spustené v WPAR majú svoj LPS rovný ich WPS.

Administrátor systému môže pomocou administračných príkazov vypísať a modifikovať rôzne množiny privilégií procesu. Príkaz **lssecattr** je možné použiť na vypísanie LPS, MPS, EPS, IPS a UPS. Príkaz **setsecattr** je možné použiť na modifikáciu LPS, MPS, EPS a IPS. UPS nie je možné modifikovať príkazom **setsecattr**, lebo UPS je atribút určený len na čítanie.

Databáza privilegovaných príkazov:

Autorizácie, roly a privilégiá umožňujú použiť odstupňované bezpečnostné prvky. Avšak skutočnosť, že RBAC využívajú rozmanité systémové operácie, umožňuje presadzovanie bezpečnostnej politiky RBAC.

Keď donedávna niektoré príkazy AIX priamo overovali autorizácie, na vykonanie takýchto kontrol bolo potrebné meniť samotný spustiteľný kód. Rozšírený režim RBAC poskytuje rámec, v ktorom sa uplatňujú kontroly autorizácií a poskytujú priradené privilégiá prostredníctvom databázy privilegovaných príkazov bez toho, aby bolo potrebné meniť systémové spustiteľné súbory.

Databáza privilegovaných príkazov poskytuje užívateľom prístup a práva k príkazom, ktoré by inak nemohli spúšťať, alebo pri ktorých by nemali potrebné privilégium na vykonanie úlohy. Databáza ukladá informácie o autorizáciách pre jednotlivé príkazy a tiež privilégiá, ktoré boli poskytnuté procesu, ak bola kontrola autorizácie úspešná. Keď je databáza uložená lokálne, nachádza sa v súbore /etc/security/privcmds a obsahuje odseky údajov vo forme atribútov príkaz verzus bezpečnosť. Nasleduje zopár kľúčových atribútov z tejto databázy (úplný popis všetkých atribútov nájdete v súbore /etc/security/privcmds).

accessauths
: Zoznam prístupových autorizácií, ktoré chránia vykonanie príkazu. Užívateľ s ktoroukoľvek z uvedených autorizácií môže spustiť príkaz a vykonať niektoré alebo všetky privilegované operácie, obsiahnuté v príkaze.

innateprivs
: Vnútorné privilégiá sú privilégiá priradené procesu, ak vyvolávač prejde kontrolou prístupových autorizácií.

authprivs
: Autorizované privilégiá sú dodatočné privilégiá priradené procesu, keď má užívateľ asociovanú autorizáciu. Tento atribút umožňuje lepšie odstupňovať riadenie príkazu a povoľuje obmedzenej množine užívateľov vykonať ďalšie privilegované operácie.

inheritprivs
: Dediteľné privilégiá sú privilégiá, ktoré proces odovzdá dcérskym procesom.

secflags
: Zoznam bezpečnostných príznakov. Príznak FSF_EPS spôsobí, že sada maximálnych privilégií (MPS) sa pri spustení príkazu načíta do sady účinných privilégií (EPS).

Keď sa užívateľ systému s rozšíreným režimom RBAC pokúsi spustiť nejaký príkaz, príkaz sa najskôr overí v databáze privilegovaných príkazov. Ak príkaz v databáze existuje, overí sa s autorizáciami asociovanými s reláciou užívateľa a s hodnotou atribútu **accessauths** pre daný príkaz. Ak relácia má jednu z uvedených autorizácií, užívateľ môže príkaz spustiť bez ohľadu na to, či užívateľ prešiel kontrolou spúšťania DAC pre príkaz. Pri vyvolaní má proces príkazu privilégiá uvedené v atribúte **innateprivs**, priradené jeho sade maximálnych privilégií (MPS). Vykonajú sa ďalšie kontroly párov autorizácia-privilégium, uvedených v atribúte **authprivs**. Ak relácia má jednu z uvedených autorizácií, asociované privilégium alebo privilégiá sa tiež pridajú do sady MPS procesu príkazu. Položka príkazu v databáze privilegovaných príkazov, ktorá má nastavenú hodnotu **FSF_EPS** v atribúte **secflags**, pri vyvolaní príkazu priradí všetky privilégiá v sade MPS do sady účinných privilégií (EPS).

Príkaz je privilegovaným príkazom, keď sa nachádza v databáze privilegovaných príkazov. Zatiaľ čo programy setuid, ktoré sa nenachádzajú v tejto databáze, sú technicky vzaté naďalej privilegovanými príkazmi, neoznačujú sa ako privilegované príkazy, keď popisujeme správanie RBAC. Ak príkaz nemá položku v databáze privilegovaných príkazov, potom nie je privilegovaným príkazom, a prístup k nemu vynucuje DAC a samotný príkaz. Navyše ak je príkaz uvedený v databáze privilegovaných príkazov, ale relácia užívateľa nemá autorizáciu, ktorá by povolila vyvolanie tohto príkazu, systém sa vráti späť ku kontrole prístupu cez DAC a ak sú kontroly úspešné, povolí spustenie príkazu.

Bolo vytvorených niekoľko riadiacich príkazov na obsluhu a dotazovanie databázy privilegovaný príkazov. Položky v databáze privilegovaných príkazov možno vytvárať a upravovať príkazom **setsecattr**, zobraziť pomocou príkazu **lssecattr** a odstrániť príkazom **rmsecattr**.

Zisťovanie vyžadovaných autorizácií pre príkaz:

Veľa systémových administračných aplikácií vyžaduje, aby sa autorizácie spúšťali korektne. Zatiaľ čo je sada preddefinovaných príkazov poskytnutá v databáze privilegovaných príkazov, administrátori systému možno budú musieť pridať položky, ktoré sú špecifické pre ich prostredie. Databáza privilegovaných príkazov umožňuje pridanie položiek do databázy. Správna autorizácia musí byť vypísaná v atribúte **accessauths**, aby mohla získať prístup k príkazu.

V operačnom systéme AIX existujú dva spôsoby použitia a kontroly autorizácie prostredníctvom rozšíreného rámca RBAC:

- Access Auths (Prístupová autorizácia): Atribút zadaný v databáze privilegovaných príkazov a obsahuje zoznam autorizačných názvov oddelených čiarkou. Užívateľ, ktorého aktuálna relácia má jednu z autorizácií v zozname, má dovolené spustiť príkaz. Kontroluje to zavádzací program systému počas spúšťania chránených privilegovaných vykonateľných súborov.
- Check Auths (checkauths()): Špecifická autorizácia alebo zoznam autorizácií možno skontrolovať programovaním pomocou aplikačného programovacieho rozhrania checkauths(). Zadané autorizácie sú skontrolované podľa autorizácií prítomných v roli v rámci aktuálnej relácie. Na základe výsledku tejto kontroly môže program vykonať privilegované operácie.

Pred pridaním príkazu do databázy privilegovaných príkazov je potrebné stanoviť sady autorizácií, aby sa zabezpečilo povolenie vykonania príkazu. Program alebo aplikácia môže interne vykonať dodatočné kontroly autorizácie. Je dôležité stanoviť zoznam autorizácií použitých v procese, ktorý môže byť priradený počas vytvárania vlastnej roly.

Nasleduje základná stratégia na stanovenie vyžadovaných autorizácií pre príkaz:

1. Prostrediu shell vyvolania priraďte privilégium **PV_ROOT** alebo priraďte rolu s autorizáciou *aix*.

   Dôležité: V globálnom oddiele WPAR musí byť privilégium **PV_ROOT** priradené do efektívnej a maximálnej sady privilégií procesu prostredia shell vyvolania. Aj v systémovom oddiele WPAR musí byť toto privilégium pridané do sady privilégií dedenia procesu.
2. Spustite príkaz.
3. Zaznamenajte autorizácie použité pre proces.
4. Autorizáciu nahlásenú pod *Access Auths* uložte do atribútu **accessauths** príkazu v databáze privilegovaných príkazov. Autorizácie nahlásené pod *Check Auths* možno použiť počas vytvárania rolí v systéme.

Tieto kroky by mali byť vykonané v riadenom prostredí, pretože privilégium **PV_ROOT** je priradené prostrediu shell alebo preberá rolu s autorizáciou *aix* a tiež preto, lebo tieto metódy sú mimoriadne výkonné. Spustenie príkazu môže mať okrem toho vplyv na systém, čo môže ovplyvniť ostatných užívateľov. V praxi pôjde pravdepodobne o metódu pokusu a omylu. Ak chcete v prípade aplikácií, ktoré sa spúšťajú na dlhý čas, získať úplnú sadu autorizácií, bude asi potrebné spustiť príkaz opakovane s rôznymi návesťami a voľbami a možno na dlhé časové obdobie. Vyžadovanú sadu autorizácií procesu možno ľahko zhromaždiť pomocou jednej z nasledujúcich procedúr, ktoré môže vykonať administrátor s náležitým oprávnením:

traceauth
: Zadajte argument, ktorý je príkazom na spustenie. Príkaz **traceauth** spustí príkaz a zaznamená obidva typy autorizácií použité počas životnosti procesu. Keď sa príkaz ukončí, príkaz **traceauth** zobrazí autorizácie, ktoré boli použité v **stdout**.

lssecattr
: Ak je príkaz procesom, ktorý sa spúšťa na dlhý čas, príkaz **lssecattr** možno použiť na zobrazenie autorizácií používaných procesom. Ak chcete v systéme zapnúť sledovanie autorizácií, spustite nasledujúci príkaz:

  setrunmode –c; setsecconf –o traceauth=enable Ak chcete zobraziť použitú autorizáciu pre proces, príkaz **lssecattr** spustite nasledovne, pričom nahraďte PID monitorovaného procesu:

  lssecattr –p –A *PID*

Ak chcete po stanovení vyžadovaných autorizácií pridať príkaz do databázy privilegovaných príkazov, vykonajte kroky v časti “Pridanie príkazu do databázy privilegovaných príkazov” na strane 93. Potom by mal príkaz spustiť autorizovaný užívateľ, aby ste si overili, že beží správne.

Určenie vyžadovaných privilégií pre príkaz:

Veľa aplikácií vyžaduje špecifické privilégiá, aby sa mohli správne spustiť. Zatiaľ čo sada preddefinovaných príkazov je poskytnutá v databáze privilegovaných príkazov, administrátori systému možno budú musieť pridať položky, ktoré sú špecifické pre ich aplikáciu alebo prostredie. Databáza privilegovaných príkazov umožňuje pridanie položiek pre príkazy a ich priradené privilégiá.

Pred pridaním príkazu do databázy privilegovaných príkazov je potrebné určiť minimálnu sadu vyžadovaných privilégií, aby bolo spustenie príkazu čo najlepšie zabezpečené. Akékoľvek privilégiá, poskytnuté nad rámec privilégií nevyhnutných pre správne spustenie, porušujú zásadu čo najnižších privilégií. Preto je pri pridávaní privilegovaného príkazu do systému veľmi dôležité určiť minimálne vyžadované privilégiá.

Nasleduje základná stratégia pre určenie minimálnych vyžadovaných privilégií pre určitý príkaz:

1. ISSO (Information System Security Officer) alebo užívateľ s rolou isso môže administrátorovi systému spúšťajúcemu príkaz, ktorý má byť priradený do privilegovanej databázy, priradiť privilégium **PV_ROOT**. Priradenie privilégia **PV_ROOT** prostrediu shell vyvolania sa vykoná pomocou príkazu setsecattr. Napríklad:

   setsecattr -p eprivs=PV_ROOT mprivs=PV_ROOT $$
2. Spustite príkaz na zhromaždenie sady privilégií.
3. Zaznamenajte sadu privilégií použitú pre proces.
4. Do atribútu **innateprivs** príkazu v databáze privilegovaných príkazov uložte potrebné privilégiá.

Tieto kroky by ste mali vykonať v kontrolovanom prostredí, pretože privilégium **PV_ROOT** je priradené prostrediu shell a má extrémne veľkú silu. Navyše, spustenie príkazu môže mať dopad na systém, ktorý sa prejaví u iných užívateľov. V praxi pôjde pravdepodobne o metódu pokusu a omylu. Aby ste získali úplnú sadu privilégií, príkaz budete pravdepodobne musieť spustiť opakovane s rôznymi príznakmi a voľbami, a v prípade dlho spustených aplikácií aj po dlhšie časové obdobie. Vyžadovaná sada privilégií procesu sa dá ľahko zhromaždiť pomocou nasledujúcich procedúr, ktoré môže vykonať administrátor s príslušným oprávnením:

tracepriv
: Vezme argument, čo je príkaz, ktorý sa má spustiť. Príkaz **tracepriv** spustí príkaz a zaznamená privilégiá počas životnosti procesu. Keď sa príkaz dokončí, príkaz **tracepriv** zobrazí privilégiá použité pri **stdout**.

lssecattr
: Ak je príkaz dlhodobo bežiaci proces, potom je možné použiť príkaz **lssecattr** na zobrazenie privilégií, ktoré proces používa. Ak chcete zobraziť sadu privilégií použitých pre určitý proces, spustite príkaz v tomto tvare, a nahraďte pritom PID monitorovaného procesu:

  lssecattr –p –a uprivs *PID*

Keď určíte minimálne vyžadované privilégiá, vykonajte kroky uvedené v časti “Pridanie príkazu do databázy privilegovaných príkazov” na strane 93 a pridajte príkaz do databázy privilegovaných príkazov. Potom by mal príkaz spustiť autorizovaný užívateľ, aby ste si overili, že beží správne.

Šírenie privilégií:

Keď systémové volanie **fork** vytvorí nový proces, **fork** mu poskytne rovnaké privilégiá, ako má rodičovský proces (proces, ktorý zavolal systémové volanie **fork**). Keď proces zavolá systémové volanie **exec** na spustiteľnom súbore, **exec** nanovo prepočíta privilégiá pre spustiteľný súbor na základe privilégií, ktoré aktuálne vlastní **exec** a privilégií, ktoré vlastní spustiteľný súbor.

Šírené privilégiá sa vypočítajú nasledovne:

1. Najskôr sa vypočíta zjednotenie (bitová operácia OR) dediteľných privilégií, ktoré vlastní starý (rodičovský) proces a množiny vlastných privilégií, ktoré vlastní spustiteľný súbor.
2. Ak je užívateľ príslušne autorizovaný, vypočíta sa zjednotenie (bitové OR) výsledku z predchádzajúceho kroku a autorizovaných privilégií.
3. Ak existujú obmedzujúce privilégiá, vypočíta sa prienik výsledku z predchádzajúceho kroku a obmedzujúcich privilégií. Obmedzujúce privilégiá, ak také sú, sa dedia cez systémové volanie **exec**.
4. Množina privilégií, ktorá je výsledkom zjednotenia, sa stane množinou maximálnych privilégií pre nový proces.
5. Ak v spustiteľnom súbore existujú zdedené privilégiá, sú priradené do množiny dediteľných privilégií v novom procese. V opačnom prípade sa množina dediteľných privilégií, ktoré vlastní starý (rodičovský) proces, prenesie do množiny dediteľných oprávnení nového procesu.

Ak má spustiteľný súbor nastavený príznak zabezpečenia súboru **FSF_EPS**, množina účinných privilégií pre nový proces bude rovnaká ako jeho množina maximálnych privilégií. V opačnom prípade budú účinné privilégiá pre nový proces rovnaké ako dediteľné privilégiá, ktoré vlastní starý (rodičovský) proces.

Pridanie príkazu do databázy privilegovaných príkazov:

Pridanie akéhokoľvek príkazu do databázy privilegovaných príkazov by ste mali dôkladne zvážiť, aby ste zabezpečili správne priradenie autorizácií a privilégií.

Úplný popis atribútov, platných pre určitý príkaz, nájdete v súbore /etc/security/privcmds. Nasledujúce otázky vám pomôžu určiť, aké položky príkaz vyžaduje:

1. Má byť prístup k spusteniu príkazu riadený autorizáciou?

   ÁNO Ak neexistuje autorizácia, vytvorte ju príkazom **mkauth**. Zadajte autorizáciu v atribúte **accessauths**.

   NIE Ak príkaz môžu spustiť všetci užívatelia, zadajte autorizáciu **ALLOW_ALL** v atribúte **accessauths**.
2. Môže vlastník príkazu alebo skupina spustiť príkaz, aj keď nemajú príslušnú autorizáciu?

   ÁNO Do zoznamu autorizácií v atribúte **accessauths** pridajte autorizáciu **ALLOW_OWNER** alebo **ALLOW_GROUP**.
3. Vyžaduje vykonanie príkazu explicitnú sadu privilégií?

   ÁNO Spustite príkaz s rôznymi voľbami ako koreňový užívateľ s príkazom **tracepriv**, aby ste určili požadované privilégiá pre atribút **innateprivs**.
4. Majú byť užívateľom so špecifickou autorizáciou poskytnuté ďalšie privilégiá?

   ÁNO Zadajte ďalší pár autorizácia-privilégium v atribúte **authprivs**.
5. Potrebuje sa príkaz správať ako program SUID alebo SGID?

   ÁNO Zadajte EUID alebo EGID podľa potreby.
6. Je potrebné, aby privilégiá priradené príkazu prešli na dcérske procesy?

   ÁNO Zadajte privilégiá v atribúte **inheritprivs**.
7. Má byť množina účinných privilégií príkazu rovná množine maximálnych privilégií v čase vyvolania programu?

   ÁNO Pre atribút secflags zadajte príznak **FSF_EPS**.

   NIE Nezadajte atribút **secflags**. Očakáva sa, že kód príkazu si zvýši alebo zníži privilégiá podľa potreby, keď príznak **FSF_EPS** nie je zadaný.
8. Musí sa príkaz spustiť so špeciálnym ID 0 skutočného užívateľa?

   ÁNO Zadajte atribút RUID.
9. Je príkaz veľmi dôležitý, je potrebné ho kontrolovať a vyžaduje prítomnosť viac ako jednej osoby pred vyvolaním?

   ÁNO Zadajte atribút **authroles** a priraďte hodnotu so zoznamom rolí. Užívatelia každej roly budú musieť byť pred spustením príkazu autentifikovaní.

Keď ste odpovedali na tieto otázky, spustite príkaz **setsecattr** s príslušnými parametrami, čím ho pridáte do databázy. Ak tento príkaz už existuje a je príkazom typu SUID alebo SGID, mali by ste zvážiť odstránenie bitov **SUID** a **SGID** zo súboru, aby bol vynútený model s najnižšími privilégiami.

Databáza privilegovaných zariadení:

Databáza privilegovaných zariadení uchováva zoznam privilégií, ktoré môžu čítať alebo zapisovať do zariadenia. Táto databáza poskytuje administrátorovi ďalší mechanizmus na riadenie prístupu k zariadeniu, ktorý môže byť riadený prostredníctvom tradičného riadenia prístupov k zariadeniu.

Keď je táto databáza uložená lokálne, nachádza sa v súbore /etc/security/privdevs. Databáza uchováva privilégiá vyžadované na prístup k určitému zariadeniu na účely čítania alebo zapisovania v týchto atribútoch:

readprivs
Vypíše privilégiá, ktoré umožňujú čítanie zo zariadenia

writeprivs
Vypíše privilégiá, ktoré umožňujú zápis do zariadenia

Keď sa vyžaduje, aby bolo privilegované zariadenie otvorené v režime čítania, otvorenie je povolené, ak v sade účinných privilégií (EPS) pre proces existuje niektoré z privilégií uvedených v atribúte **readprivs**. Podobne, ak je zariadenie otvorené v režime zápisu, v EPS musí existovať privilégium v atribúte **writeprivs**.

Proces pridania nejakého zariadenia do databázy privilegovaných zariadení sa obvykle nepovažuje za bežnú operáciu. Na vypísanie a obsluhu databázy môžete použiť príkazy **lssecattr** a **setsecattr**, no pridanie alebo zmena položiek v databáze si vyžaduje významné skúmanie. Keďže oprávnenie na čítanie a zápis pre určité zariadenie je riadené prostredníctvom privilégií, musíte veľmi dôkladne preskúmať, ktoré príkazy a aplikácie potrebujú príkaz k danému zariadeniu, aby boli zadané správne privilégiá.

Databáza privilegovaných súborov:

Mnohé konfiguračné súbory systému v tradičných systémoch UNIX sú vo vlastníctve užívateľa root a ostatní užívatelia ich nemôžu priamo modifikovať. RBAC umožňuje užívateľovi modifikovať tieto konfiguračné súbory systému aktivovaním roly a spustením príkazu na získanie privilégií potrebných na modifikáciu súboru.

Sú určité konfiguračné súbory AIX, ktoré nemajú príkazové rozhrania umožňujúce modifikáciu súboru. V takýchto prípadoch je potrebné mať nástroj, ktorý umožňuje administrátorovi s príslušnou autorizáciou priamo upravovať a uložiť súbor, ku ktorému by inak nemal prístup.

Databáza privilegovaných súborov poskytuje na určovanie prístupu ku konfiguračným súborom systému pomocou autorizácií. Keď je databáza uložená lokálne, nachádza sa v súbore /etc/security/privfiles. Táto databáza namapuje konfiguračné súbory na autorizácie vyžadované na zobrazovanie alebo modifikáciu týchto súborov. Prístup ku konfiguračnému súboru je ovládaný v tejto databáze pomocou nasledujúcich atribútov:

readauths
Zoznam autorizácií s povolením čítať zo súboru

writeauths
Zoznam autorizácií s povolením zapisovať do súboru (v tomto prípade je zahrnutá aj autorizácia na čítanie)

Položky v databáze privilegovaných súborov možno vypísať príkazom **lssecattr** a vytvoriť alebo modifikovať príkazom **setsecattr**. K súborom definovaným v databázi privilegovaných súborov majú prístup autorizovaní užívatelia pomocou príkazu **/usr/bin/pvi**. Príkaz **pvi** je privilegovanou a obmedzenou verziou editora **vi** založený na príkaze **/usr/bin/tvi**. Príkaz **pvi** zavádza všetky tie isté bezpečnostné opatrenia ako príkaz **tvi** (napríklad žiadne príznaky **–r** alebo **-t**, žiadne úniky z prostredia shell, žiadne užívateľom definované makrá) a tiež vynucuje nasledujúce obmedzenia:

- Systém musí byť v rozšírenom režime RBAC.
- Otvoriť je možné len súbory definované v databázi privilegovaných súborov.
- Súčasne je možné otvoriť len jeden súbor.
- Zapisovanie do iného názvu súboru, ako je zadaný v príkazovom riadku, je zablokované.
- Súbor /etc/security/privfiles nie je možné upravovať pomocou príkazu **pvi**.
- Pokusy o otvorenie odkazov zlyhajú. Možné je upravovať len bežné súbory.

Kontroly autorizácie sa vykonajú pred otvorením súboru. Ak sa autorizácia zhoduje, množina privilégií procesu sa zvýši tak, aby zahrňovalo **PV_DAC_R** alebo **PV_DAC_W** (v závislosti od toho, či je súbor otváraný na čítanie alebo zapisovanie). Ak sa autorizácia nezhoduje, zobrazí sa chybová správa a užívateľovi sa zamedzí prístup k súboru pomocou príkazu **pvi**.

Bezpečnostné tabuľky kernelu (KST):

Informácie obsiahnuté v databáze autorizácií, rolí, privilegovaných príkazov a privilegovaných zariadení sa nepoužívajú pri zvažovaní bezpečnosti, pokiaľ údaje neboli odoslané do oblasti kernelu, ktorá sa označuje ako tabuľky Kernel Security Tables (KST). V rozšírenom režime RBAC sa kontrola autorizácie a privilégií vykonáva v kerneli, takže databázy musia byť poslané do kernelu, aby mohli byť použité.

Tabuľky KST tvoria tieto podtabuľky:

- Kernelová tabuľka autorizácií (KAT)
- Kernelová tabuľka rolí (KRT)
- Kernelová tabuľka príkazov (KCT)
- Kernelová tabuľka zariadení (KDT)

Všetky tieto tabuľky alebo vybraté tabuľky môžu byť odoslané do kernelu z priestoru užívateľa pomocou príkazu **setkst**. Tabuľky KRT a KCT sú závislé od KAT, preto keď sa aktualizuje KAT, zaktualizujú sa aj KRT a KCT, aby boli zosynchronizované. Preferovaná metóda pridania aktualizácií do tabuliek KST je vytvoriť alebo modifikovať všetky potrebné databázy na úrovni užívateľa (napríklad pomocou príkazov **mkauth**, **chauth**, **mkrole** a **setsecattr**) a potom pomocou príkazu **setkst** odoslať tabuľky do kernelu. Akonáhle budú načítané do kernelu, môžete použiť príkaz **lskst** a zobraziť informácie v jednotlivých tabuľkách.

Určitá tabuľka z bezpečnostných tabuliek kernelu sa vždy odosiela ako úplná tabuľka. Inými slovami, tabuľky KST nepovoľujú modifikácie jednotlivých položiek; je potrebné nahradiť celú tabuľku. Pred odoslaním tabuliek do kernelu príkaz **setkst** overuje platnosť tabuliek a vzťahy medzi nimi. Príkaz **setkst** je tiež umiestnený v súbore inittab, čo zaručuje, že databázy budú odoslané do KST v ranom štádiu procesu zavedenia.

Ak z nejakého dôvodu nie je možné tabuľky vytvoriť, alebo sa nedajú odoslať do kernelu a predtým neboli žiadne tabuľky načítané, systém bude fungovať, akoby neexistovali žiadne autorizácie ani roly. Príkazy, API a systémové volania na kontrolu autorizácie alebo roly v tomto scenári vrátia zlyhanie, pretože sa nenájde žiadna zhoda. Funkčnosť systému v tomto stave je podobný jeho funkčnosti v režime RBAC, až na to, že užívatelia nemôžu pristupovať k častiam kódu, ktoré slúžia na uplatnenie autorizácií.

Zakázanie koreňového užívateľa:

V rozšírenom režime RBAC sa dá systém nakonfigurovať tak, aby koreňový užívateľ nemal priradené žiadne špeciálne práva a aby k nemu systém pristupoval ako k normálnemu užívateľovi.

Už dávnejšie operačný systém chápe hodnotu 0 pre ID koreňového užívateľa ako privilegované ID, takže mu umožňuje obísť inak nutné bezpečnostné kontroly. Zakázanie koreňového užívateľa účinne odstráni kontroly v operačnom systéme, ktoré dovoľujú užívateľovi s ID = 0 obísť bezpečnostné kontroly a namiesto toho vyžaduje, aby mal proces potrebné privilégiá na prechod bezpečnostnou kontrolou. Zakázanie koreňového užívateľa minimalizuje škody spôsobené potenciálnym útočníkom, pretože v systéme už viac neexistuje žiadna identita užívateľa so všemocnými právami. Keď je koreňový užívateľ zakázaný, správu systému musia vykonávať užívatelia, ktorí majú priradené privilegované roly.

Koreňové práva môžete zakázať príkazom **/usr/sbin/setsecconf**. Spustite nasledujúci príkaz a rebootujte systém, aby ste zakázali práva koreňového užívateľa:

```
setsecconf –o root=disable
```

Po spustení tohto príkazu sa ku koreňovému užívateľskému kontu nedá pristúpiť prostredníctvom vzdialeného ani lokálneho prihlasovacieho mena ani prostredníctvom príkazu su. Keďže však koreňové užívateľské konto zostáva vlastníkom súborov v súborovom systéme, ak nejaký užívateľ nadobudne toto konto, získa tým prístup k privilegovaným súborom.

Na systéme, kde bol koreňový užívateľ zakázaný, nie sú procesom, ktoré vlastní, priradené žiadne špeciálne práva ani privilégiá. Mali by ste to vziať do úvahy, ak koreňový užívateľ na systéme vlastní aplikácie setuid, ktoré neboli pridané

do databázy privilegovaných príkazov. Takéto aplikácie setuid pravdepodobne zlyhajú v prostredí so zakázaným koreňovým užívateľom, pretože proces nebude schopný vykonávať privilegované operácie. V systéme so zakázaným koreňovým užívateľom by sa mali všetky príkazy, ktoré potrebujú vykonávať privilegované operácie, pridať do databázy privilegovaných príkazov a mali by im priradené príslušné privilégiá. Preto skôr ako zakážete práva koreňového užívateľa, by ste najskôr mali vykonať dôkladnú analýzu systému a v ňom používaných aplikácií.

Podpora vzdialenej databázy RBAC:

V podnikovom prostredí je potrebná schopnosť implementovať a presadiť spoločnú bezpečnostnú politiku na všetkých systémoch v prostredí. Ak databázy, ktoré riadia politiku, sú uložené nezávisle na každom systéme, správa bezpečnostnej politiky sa stane príťažou pre určeného administrátora systému. Rozšírený režim RBAC systému AIX umožňuje uloženie databáz RBAC na LDAP, takže bezpečnostná politika pre všetky systémy v prostredí sa dá riadiť centrálne.

V systéme AIX bola pridaná podpora pre všetky databázy týkajúce sa RBAC na ich ukladanie na LDAP. Nasledujú príslušné databázy RBAC:

- Databáza autorizácií
- Databáza rolí
- Databáza privilegovaných príkazov
- Databáza privilegovaných zariadení
- Databáza privilegovaných súborov

Poznámka: Databáza autorizácií uložená na LDAP obsahuje len užívateľom definované autorizácie. Systémom definované autorizácie nie je možné ukladať na LDAP a pre každý klientsky systém zostávajú lokálne.

AIX poskytuje pomocné programy na jednoduchý export údajov lokálneho RBAC na LDAP, na konfigurovanie klienta na používanie údajov RBAC v LDAP, na radenie vyhľadávania údajov RBAC a na správu údajov LDAP z klientskeho systému. Nasledujúce časti obsahujú ďalšie informácie o funkciách LDAP, ktoré sú dostupné v rozšírenom RBAC.

Export údajov RBAC do LDAP:

Úvodná príprava na použitie LDAP ako archívu databáz RBAC vyžaduje zaplnenie LDAP servera údajmi RBAC.

LDAP server musí mať najskôr nainštalovanú schému RBAC pre LDAP, až potom môžu klienti LDAP použiť server pre údaje RBAC. Schéma RBAC pre LDAP je dostupná na systéme AIX v súbore /etc/security/ldap/sec.ldif . Schému LDAP servera je potrebné aktualizovať týmto súborom pomocou príkazu **ldapmodify**.

Súbor **/usr/sbin/rbactoldif** môžete použiť na čítanie údajov v lokálnych databázach RBAC a potom ich posunúť ďalej vo formáte vhodnom pre LDAP. Výstup, ktorý vygeneruje príkaz **rbactoldif**, môžete uložiť do súboru a následne ho použiť na zaplnenie servera LDAP údajmi pomocou príkazu **ldapadd**. Príkaz **rbactoldif** použije nasledujúce databázy na lokálnom systéme na vygenerovanie údajov RBAC pre LDAP:

- /etc/security/authorizations
- /etc/security/privcmds
- /etc/security/privdevs
- /etc/security/privfiles
- /etc/security/roles

Umiestnenie pre úložný priestor LDAP pre údaje RBAC by ste mali dobre zvážiť. Odporúča sa, aby boli údaje RBAC v LDAP umiestnené pod rovnakým rodičovským DN ako užívateľské a skupinové údaje. Zoznamy ACL k týmto údajom by ste potom mali podľa potreby prispôsobiť zvolenej bezpečnostnej politike.

Konfigurácia klienta LDAP pre RBAC:

Systém musí byť nakonfigurovaný ako klient LDAP, aby mohol používať úaje RBAC uložené v adresári LDAP.

Môžete použiť príkaz AIX **/usr/sbin/mksecldap** a nakonfigurovať systém ako klienta LDAP. Príkaz **mksecldap** dynamicky prehľadá zadaný LDAP server, aby určil umiestnenie údajov ohľadom autorizácií, rolí, privilegovaných príkazov, zariadení a súborov, a výsledok uloží do súboru /etc/security/ldap/ldap.cfg.

Po úspešnom nakonfigurovaní systému ako klienta LDAP pomocou príkazu **mksecldap** je ďalej potrebné nakonfigurovať systém tak, aby bol LDAP povolený ako doména pre hľadanie údajov RBAC. Súbor /etc/nscontrol.conf treba upraviť tak, aby obsahoval LDAP v atribúte **secorder** pre databázy uložené v LDAP.

Akonáhle je systém nakonfigurovaný ako klient LDAP aj ako vyhľadávacia doména pre údaje RBAC, klientsky démon **/usr/sbin/secldapclntd** bude periodicky získavať údaje RBAC z LDAP a odosielať ich do bezpečnostných tabuliek kernelu (KST) prostredníctvom príkazu **setkst**. Môžete nakonfigurovať periódu, v ktorej má démon získavať údaje RBAC z LDAP, a to pomocou atribútu **rbacinterval** v súbore /etc/security/ldap/ldap.cfg. Predvolená hodnota tohto atribútu je 3600, čo znamená, že každú hodinu sa získajú údaje RBAC z LDAP a budú aktualizované tabuľky KST. Tabuľky KST môžu byť aktualizované aj ručne, keď administrátor spustí príkaz **setkst**.

Riadiaci súbor názvovej služby:

Údaje RBAC môžu byť trvalo umiestnené iba v lokálnych súboroch, iba v LDAP, alebo môžu byť rozptýlené v lokálnych súboroch aj LDAP, a to nakonfigurovaním danej databázy v riadiacom súbore názvovej služby /etc/nscontrol.conf.

Poradie vyhľadávania pre databázu autorizácií, rolí, privilegovaných príkazov, zariadení a súborov je určené individuálne v súbore /etc/nscontrol.conf. Poradie vyhľadávania pre určitú databázu je určené v súbore prostredníctvom atribútu **secorder**, čo je vlastne zoznam domén oddelených čiarkami. Nasleduje ukážka konfigurácie pre databázu autorizácií:

```
authorizations:
        secorder = LDAP,files
```

Tento príklad určuje, že dotazy ohľadom autorizácií sa budú najskôr hľadať v LDAP a ak sa tam nenájdu, v lokálnych súboroch. Kolekcia autorizácií, dostupných na systéme, je kombináciou autorizácií z LDAP a autorizácií z lokálnych súborov. Nejde len o jednoduchú kombináciu hodnôt z týchto dvoch domén, ale skôr o zjednotenie hodnôt. V prípade konfigurácie, uvedenej vyššie, boli zahrnuté všetky autorizácie z LDAP, a k výsledku boli zahrnuté iba jedinečné autorizácie z lokálnych súborov.

Pokus o zmenu a vymazanie sa vykoná na prvej uvedenej doméne, a iba ak sa tam požadovaná entita nenachádza, prídu na rad ďalšie domény. V danom prípade sa najskôr prehľadá LDAP, a ak sa tam autorizácia nenájde, prehľadajú sa aj lokálne súbory. Nové položky sa vždy vytvárajú v doméne, ktorá je v atribúte **secorder** uvedená ako prvá. V uvedenom príklade sa pokus o vytvorenie novej autorizácie vykoná v databáze LDAP.

Ak sa v súbore /etc/nscontrol.conf pre nejakú databázu nenachádza žiadna entita, alebo ak súbor neexistuje, dotazy a zmeny týkajúce sa databázy sa vykonajú iba v databáze lokálnych súborov. Konfiguráciu pre databázu v súbore môžete nastaviť príkazom **chsec** a vypísať pomocou príkazu **lssec**. Ak chcete nakonfigurovať, aby sa údaje o autorizáciách získavali najskôr z LDAP a potom z lokálnych súborov, spustite nasledujúci príkaz:

```
chsec –f /etc/nscontrol.conf –s authorizations –a secorder=LDAP,files
```

Konfigurácia v súbore /etc/nscontrol.conf riadi knižničné rozhranie aj rozhranie príkazového riadka. Aplikácie môžu získať aktuálnu hodnotu atribútu **secorder** pre databázu prostredníctvom rozhrania **getsecorder**. Hodnota atribútu **secorder** pre proces môže byť prepísaná pomocou rozhrania **setsecorder**.

Povolenie príkazov RBAC pre LDAP:

Všetky príkazy na správu databázy RBAC majú povolené používať konfiguráciu v súbore /etc/nscontrol.conf a dotazovať, modifikovať, vytvárať alebo odstraňovať entitu v doméne alebo doménach zadefinovaných pre danú databázu.

Štandardne sa domény spracúvajú, ako definuje atribút **secorder** pre databázu, toto nastavenie je však možné obísť použitím voľby **–R** v príkazovom riadku. Zadanie voľby **–R** pre príkaz vynúti vykonanie operácie na zadanej doméne, a prepíše konfiguráciu v súbore /etc/nscontrol.conf. Pre podporu vzdialených domén sú k dispozícii nasledujúce príkazy na správu databáz RBAC:

- **mkauth**, **chauth**, **lsauth** a **rmauth**
- **mkrole**, **chrole**, **lsrole** a **rmrole**
- **setsecattr**, **lssecattr** a **rmsecattr**

Okrem toho, príkaz **setkst** je povolený na použitie konfigurácie zo súboru /etc/nscontrol.conf. Príkaz **setkst** získa zlúčenú kópiu položiek pre danú databázu, ako je zadefinované v súbore, a výsledné údaje načíta do bezpečnostných tabuliek kernelu.

Priradenie naprieč doménami:

Keď navrhujete prostredie, kde údaje RBAC poskytujú dve domény, napríklad lokálne súbory a LDAP, musíte dôkladne uvážiť otázku priraďovania entít medzi doménami. Príkladom priraďovania naprieč doménami je priradenie roly definovanej v LDAP lokálnemu užívateľovi alebo priradenie lokálne zadefinovanej roly užívateľovi LDAP.

Priradenie vzdialenej entity (LDAP roly) lokálnej entite (lokálnemu užívateľovi) nie je žiaden problém, keďže nemá dopad na iné systémy v danom prostredí. Avšak priradenie lokálnej entity (lokálnej roly) vzdialenej entite (užívateľovi LDAP) musíte robiť veľmi opatrne. Keďže vzdialená entita (užívateľ LDAP) je viditeľná na viacerých klientoch, nedá sa zaručiť, že lokálna entita (lokálna rola) k nej priradená je zadefinovaná, respektíve že má rovnakú definíciu na každom klientskom systéme. Napríklad na každom klientovi môže byť určitá rola lokálne definovaná, má však priradené rozličné autorizácie. Vzdialený užívateľ, ktorý má priradenú túto lokálnu rolu, preto bude mať na jednotlivých klientoch rozdielne autorizácie, čo môže mať neželané dôsledky pre bezpečnosť.

Aby ste zabránili potenciálnym bezpečnostným problémom v súvislosti s priradením lokálnej entity entite LDAP, odporúča sa, aby LDAP server implementoval riadenie prístupov k databáze RBAC, čo užívateľom zabráni modifikovať položky. Modifikovať entity LDAP RBAC by malo byť povolené iba užívateľom, ktorí sa k LDAP serveru pripájajú prostredníctvom privilegovaného konta. Ostatní klienti by mali mať iba oprávnenie na čítanie databáz LDAP RBAC.

Veľkostný limit v rozšírenom RBAC:

Nasledujúca tabuľka uvádza rôzne limity pre elementy súvisiace s RBAC:

Tabuľka 10. rôzne limity pre elementy súvisiace s RBAC

Popis	Maximálna veľkosť
Názov roly	63 tlačiteľných znakov
Maximálny počet rolí na reláciu	8
Maximálna veľkosť názvu autorizácie	63 tlačiteľných znakov
Maximálny počet úrovní v hierarchii autorizácií	9
Maximálny počet prístupových autorizácií na príkaz	8
Maximálny počet sád autorizovaných privilégií na príkaz	8

Správa rozšíreného RBAC:

Táto časť opisuje bežné scenáre použitia príkazového riadka na správu RBAC. Tieto príklady ilustrujú hlavné aspekty fungovania tejto funkcie. Pre správu RBAC sú tiež k dispozícii rozhrania SMIT. Rýchla cesta k ponukám RBAC rozhrania SMIT je smit rbac.

Vytvorenie užívateľom definovanej autorizácie:

Môžete vytvárať užívateľom definované autorizácie, ktoré možno používať na riadenie vykonávania príkazov.

Užívateľom definované autorizácie vytvoríte pomocou príkazu mkauth. Zmeny v databázi autorizácií nadobudnú platnosť po prevzatí týchto zmien do jadra príkazom **setkst**.

- Na vytvorenie užívateľom definovanej autorizácie použite nasledujúci príkaz:
  mkauth *názov_autor.*

Vytváranie a modifikovanie rolí:

Rolu môžete vytvoriť príkazom **mkrole**.

Rolu sa vytvárajú príkazom **mkrole**. Zmeny v databáze rolí nadobudnú účinnosť, keď sa príkazom **setkst** načítajú do kernelu. Roly môžete modifikovať príkazom **chrole**.

- Spustením nasledujúceho príkazu vytvorte rolu:
  mkrole dflt_msg="*My Role*" *role_name*
- Ak chcete vytvoriť rolu a zdediť autorizácie od existujúcich rolí, spustite nasledujúci príkaz:
  mkrole rolelist=*child_role1***,***child_role2 role_name*
- Ak chcete modifikovať definíciu roly, spustite nasledujúci príkaz:
  chrole rolelist=*child_role3 role_name*

Priraďovanie autorizácií roliam:

Autorizácie môžete priradiť roli pomocou príkazov **mkrole** a **chrole**.

- Spustením príkazu **mkrole** priradíte autorizácie **názov_autor.1** a **názov_autor.2** roli **názov_role**:
  `mkrole authorizations=názov_autor.1,názov_autor.2 názov_role`
- Spustením príkazu **chrole** priradíte autorizácie **názov_autor.1** a **názov_autor.2** roli **názov_role**:
  `chrole authorizations=názov_autor.1,názov_autor.2 názov_role`

Nastavenie režimu autentifikácie pre rolu:

Môžete riadiť aktivovanie rolí pomocou atribútu roly **auth_mode**.

Platné hodnoty pre atribút **auth_mode**:

NONE Autentifikácia nie je potrebná

INVOKER
Vyvolávač musí najskôr zadať svoje heslo. Ide o predvolenú hodnotu.

Zadajte nasledujúci príkaz, ktorý užívateľov prinúti autentifikovať sa ako oni sami, keď nadobúdajú danú rolu:

chrole auth_mod=INVOKER *role_name*

Priradenie rolí užívateľom:

Pomocou príkazu **chuser** môžete užívateľom priradiť roly.

Spustením nasledujúceho príkazu priraďte roly **role_name1** a **role_name2** užívateľovi **user_name**:
`chuser roles=role_name1,role_name2 user_name`

Aktivovanie rolí:

Štandardne musí užívateľ aktivovať rolu v relácii, aby mohol spúšťať privilegované príkazy.

- Roly **role_name1** a **role_name2** aktivujete spustením nasledujúceho príkazu:
  `swrole role_name1,role_name2`

- Niektoré roly, priradené užívateľom, sú klasifikované ako predvolené roly. Tieto roly sa aktivujú automaticky po prihlásení užívateľa. Ostávajú aktívne počas celej prihlasovacej relácie. Ak chcete užívateľovi priradiť rolu role_name1 ako predvolenú rolu, spustite nasledujúci príkaz:

  ```
  chuser roles=role_name1,role_name2 default_roles=role_name1 user_name
  ```

Vypísanie sady aktívnych rolí:

Pomocou príkazu **rolelist** s voľbou **-e** môžete zobraziť informácie o sade účinných aktívnych rolí pre určitú reláciu.

- Ak chcete zobraziť sadu účinných aktívnych rolí pre určitú reláciu, spustite nasledujúci príkaz:

  ```
  rolelist -e
  ```

Vypísanie rolí pre užívateľa:

Príkaz **rolelist** poskytuje informácie o roli a autorizácii k aktuálnym rolám užívateľa alebo k rolám, ktoré mu boli priradené.

Štandardne príkaz **rolelist** zobrazí zoznam rolí, ktoré boli priradené užívateľovi. Rovnaké informácie v zásade zobrazí aj príkaz lsuser -a roles user1, až na to, že vypíše tiež popis roly, ak je k dispozícii.

- Ak chcete vypísať vám priradené roly a autorizácie, spustite nasledujúci príkaz:

  ```
  rolelist -a
  ```

Audit rolí relácie:

Roly, ktoré sú aktívne v prihlasovacej relácii, sú auditované spolu s ďalšími atribútmi typu UID a GID. Tieto roly môžete vypísať príkazom **auditpr**.

Ak chcete zobraziť roly z auditovacej stopy, spustite nasledujúci príkaz:

```
auditpr -h eli -i /audit/trail
```

Priradenie privilégií spustenému procesu:

Pomocou príkazu **setsecattr** môžete modifikovať privilégiá spusteného procesu.

- Ak chcete aktualizovať sadu efektívnych privilégií pre proces, zadajte nasledujúci príkaz:

  setsecattr –p eprivs=*privileges pid*
- Skôr ako pridáte akékoľvek privilégium do sady účinných privilégií procesu sa uistite, že toto privilégium už existuje v sade maximálnych privilégií. Ak chcete modifikovať sadu maximálnych privilégií, spustite nasledujúci príkaz:

  setsecattr –p mprivs=*privileges pid*

Správa privilégií WPAR:

Každému WPAR je priradená sada privilégií, ktorá určuje jeho právomoci. Označuje sa ako sada privilégií WPAR (WPS).

Procesy spustené vo WPAR môžu použiť iba privilégiá, ktoré sú dostupné vo WPS.

- Ak chcete modifikovať WPS globálneho WPAR, spustite nasledujúci príkaz:

  chwpar –S privs+=*privileges wpar_name*

Určenie privilégií vyžadovaných pre príkaz:

Niektoré príkazy vyžadujú špeciálne privilégiá na vykonávanie privilegovaných operácií. Privilégiá sa používajú v kerneli, aby obchádzali bezpečnostné obmedzenia.

Pomocou príkazu **tracepriv** môžete profilovať príkaz, aby ste určili privilégiá, ktoré sa vyžadujú na úspešné spustenie daného príkazu. Príkaz **tracepriv** zaznamenáva privilégiá, ktoré použil iný príkaz, keď bol spustený. Príkaz by mal byť spustený s privilégiom **PV_ROOT**, aby boli úspešné všetky pokusy použiť privilégiá. Keď sa príkaz dokončí, sada použitých privilégií sa odošle do stdout.

- Ak chcete profilovať daný príkaz, spustite príkaz:
  tracepriv –ef *command_name*

Používanie autorizácii na riadenie príkazov:

Autorizácie je možné používať na riadenie spúšťania príkazov.

Pomocou príkazu **setsecattr** môžete priradiť autorizácie k príkazu. Príkaz **setsecattr** pridá strofu k databáze privilegovaných príkazov (/etc/security/privcmds). Modifikácie v tejto databáze sa musia prevziať do jadra príkazom **setkst**.

- Ak chcete priradiť autorizácie k príkazu, spustite nasledujúci príkaz:
  setsecattr –c accessauths=*názvy_autor.* **innateprivs=***privilégiá* **proxyprivs=***privilégiá* **authprivs=***názov_autor.***=***názov_príkazu privilégií*

Riadenie prístupu k zariadeniam:

RBAC poskytuje mechanizmus na ďalšie riadenie prístupu k zariadeniam. Administrátor systému môže zadávať privilégiá potrebné na otvorenie zariadenia v režime čítania alebo zapisovania.

Napríklad oprávnenie na zápis pre DVD napaľovačku môže riadiť privilégium **PV_DEV_CONFIG**, takže DVD disky budú môcť vytvárať iba procesy, ktoré majú toto privilégium.

- Ak chcete pridať zariadenie do databázy zariadení, spustite nasledujúci príkaz:
  setsecattr –d readprivs=*privileges* **writeprivs=***privileges device_name*

Aktualizácia bezpečnostných tabuliek jadra RBAC:

Príkaz **setkst** prečíta bezpečnostné databázy a načíta informácie z týchto databáz do bezpečnostných tabuliek jadra (KST).

Štandardne sa všetky bezpečnostné databázy posielajú do KST. Alebo je možné zadať konkrétnu databázu pomocou voľby **-t**. Avšak pri zadaní, že do KST sa má odoslať len databáza autorizácií, sa tiež aktualizuje databáza rolí a privilegovaných príkazov v KST, lebo databáza rolí a privilegovaných príkazov je závislá od databáze autorizácií.

- Ak chcete odoslať do jadra všetky najnovšie databázy RBAC, spustite nasledujúci príkaz:
  ```
  setkst
  ```

Použitie prepínača rozšíreného režimu RBAC:

Prepínač celosystémovej konfigurácie umožňuje zakázať schopnosti rozšíreného RBAC a vrátiť sa späť k predchádzajúcemu správaniu RBAC.

Administrátor systému môže zakázať rozšírený režim RBAC spustením príkazu **chdev** na zariadení **sys0** a zadaním atribútu **enhanced_RBAC** s hodnotou **false**, na čo rebootuje systém. Režim sa dá prepnúť späť do rozšíreného režimu RBAC nastavením atribútu **enhanced_RBAC** na hodnotu **true** a následným opätovným zavedením systému.

- Ak sa chcete vrátiť k predchádzajúcemu, klasickému režimu RBAC, spustite nasledujúci príkaz:
  ```
  chdev -l sys0 -a enhanced_RBAC=false
  ```
- Ak chcete vypísať hodnotu atribútu **enhanced_RBAC**, spustite nasledujúci príkaz:
  ```
  lsattr -E -l sys0 -a enhanced_RBAC
  ```

V prostredí WPAR sa dá režim RBAC konfigurovať iba z globálneho systému, a ovplyvní globálny systém aj všetky WPAR.

Poznámka: Zakázanie rozšíreného režimu RBAC môže znížiť bezpečnostný prah vášho systému, najmä vo WPAR.

Príkazy týkajúce sa RBAC

Nasledujúca tabuľka uvádza zoznam príkazov súvisiacich s RBAC, ktoré sú v operačnom systéme AIX k dispozícii na správu a využívanie rámca RBAC.

Command	Popis
chauth	Upraviť atribúty užívateľom definovanej autorizácie
chrole	Upraviť atribúty roly
ckauth	Skontrolovať autorizáciu pre aktuálny proces
lsauth	Zobraziť atribúty užívateľom a systémom definovanej autorizácie
lskst	Vypísať položky v bezpečnostných tabuľkách jadra
lspriv	Zobraziť privilégiá dostupné v systéme
lsrole	Zobraziť atribúty roly
lssecattr	Zobraziť bezpečnostné atribúty príkazu, zariadenia, procesu alebo súboru
mkauth	Vytvoriť novú užívateľom definovanú autorizáciu
mkrole	Vytvoriť novú rolu
pvi	Privilegovaný súborový editor
rbacqry	Povoľuje aplikácie RBAC
rbactoldif	Vypísať databázy RBAC užívateľskej úrovne vo formáte kompatibilnom s LDAP
rmauth	Odstrániť užívateľom definované autorizácie
rmrole	Odstrániť rolu
rmsecattr	Odstrániť definíciu bezpečnostných atribútov pre príkaz, zariadenie alebo súbor
rolelist	Zobraziť informácie o roli pre užívateľa alebo proces
setkst	Odoslať položky v databázach RBAC užívateľskej úrovne do bezpečnostných tabuliek jadra
setsecattr	Nastaviť bezpečnostné atribúty príkazu, zariadenia, procesu alebo súboru
setsecconf	Upraviť bezpečnostné príznaky jadra
swrole	Vytvoriť novú reláciu roly
tracepriv	Sledovať privilégiá potrebné na úspešné vykonanie príkazu

Súbory týkajúce sa RBAC

Nasledujúca tabuľka obsahuje súbory týkajúce RBAC v systémeAIX na konfigurovanie a ukladanie databázových informácií.

Súbor	Popis
/etc/nscontrol.conf	Riadiaci súbor názvovej služby pre určité bezpečnostné databázy
/etc/security/authorizations	Databáza užívateľom definovaných autorizácií
/etc/security/privcmds	Databáza privilegovaných príkazov
/etc/security/privfiles	Databáza privilegovaných súborov
/etc/security/privdevs	Databáza privilegovaných zariadení

Súbor	Popis
/etc/security/roles	Databáza rolí

Používanie rozšíreného RBAC v aplikáciách

Mnoho aplikácií nevyžaduje žiadne úpravy, aby mohli byť úspešne spúšťané v rozšírenom prostredí RBAC. Stačí len zadefinovať prístupové autorizácie aplikácie a priradené privilégiá a potom priradiť aplikáciu k databáze privilegovaných príkazov.

Aplikácia však môže používať rozšírené RBAC volaním rozhraní RBAC na riadenie vykonávania aplikácie na granulovanej úrovni, čo má potom za následok bezpečnejšiu aplikáciu. K aplikáciám, ktoré by mohli mať úžitok z integrácie s rozšíreným RBAC, patria:

- Aplikácie, ktoré obmedzujú používanie buď na užívateľa root alebo členov určitej skupiny. Tieto aplikácie väčšinou kontrolujú účinnú identitu užívateľa alebo členstvo v skupine a je možné ich upraviť tak, aby namiesto toho kontrolovali autorizáciu.
- Aplikácie, ktoré pomocou bitov režimu **setuid** alebo **setgid** povoľujú neprivilegovaným užívateľom získavať privilégiá počas vyvolania príkazu. Tieto aplikácie by zvyčajne boli bezpečnejšie, keby používali rámovanie privilégií, takže na vykonanie ich úlohy by sa použilo menšie privilégium.

Kontrola autorizácie:

Aplikácie, ktoré aktuálne používajú ID užívateľa alebo ID skupiny volajúceho užívateľa na zistenie, či je schopný vykonávať privilegované operácie, by mali byť modifikované, aby namiesto toho kontrolovali autorizáciu.

Vezmime si napríklad aplikáciu, ktorá vykonáva úlohy konfigurácie systémových súborov a aktuálne dovoľuje koreňovému užívateľovi (UID = 0) vykonávať niektoré privilegované operácie:

```
if (getuid() == 0) {
   /* dovoliť pokračovať v privilegovaných operáciách */
}
```

Namiesto toho, aby aplikácia dovoľovala užívateľom so špecifickou autorizáciou (**aix.fs.config**) vykonávať privilegované operácie, môžete kód zmeniť tak, aby sa na kontrolu autorizácie použilo API rozhranie **checkauths**:

```
if (checkauths("aix.fs.config", CHECK_ALL)) {
  /* dovoliť pokračovať v privilegovaných operáciách */
 }
```

API rozhranie **checkauths** je povolené pre klasický aj rozšírený režim RBAC a vráti kód úspechu **0**, ak volajúci proces má špecifikovanú autorizáciu. API rozhranie **checkauths** tiež určuje, či sú práva koreňového užívateľa povolené alebo zakázané, a potom umožní alebo zakáže koreňovému užívateľovi obísť kontrolu autorizácie. Pred verziou systému AIX verzie 6.1 sa rozhrania API **MatchAllAuths**, **MatchAnyAuths**, **MatchAllAuthsList** a **MatchAnyAuthsList** normálne používali na kontrolu autorizácie. Aplikácie poskytnuté v systéme AIX verzie 6.1 a jeho neskorších verziách by mali namiesto nich používať API rozhranie **checkauths**, pretože podporuje klasický aj rozšírený režim RBAC a zakázanie koreňového užívateľa.

Podobne ako v uvedenom príklade, aplikácie, ktoré volajú **getuid**, **getgid** alebo podobnú funkciu len na to, aby určitým užívateľom dovolili vykonávať špecifické úlohy, je možné modifikovať tak, aby na kontrolu autorizácie namiesto toho používali API rozhranie **checkauths**. Ak kontrolované ID užívateľa alebo ID skupiny nepatrí koreňovému užívateľovi, môžete najskôr použiť systémové volanie **sys_parm** a zistiť, či je rozšírené RBAC povolené alebo nie. Ak rozšírené RBAC nie je povolené, kód môže vykonať už prebiehajúce kontroly. V opačnom prípade, ak je rozšírené RBAC povolené, kód môže skontrolovať relevantné systémové alebo užívateľom definované autorizácie.

Prechodné zvýšenie privilégií:

Keď raz boli aplikácie modifikované tak, aby kontrolovali autorizácie, môžete ich ďalej zmeniť tak, aby využívali jemne odstupňované prechodné zvýšenie privilégií počas chodu.

Aplikácie môžu použiť API rozhranie **priv_raise** na zvýšenie privilégií potrebných na vykonanie operácie, a potom znížiť privilégiá cez API rozhranie **priv_lower**. Zvýšenie privilégií tesne pred pokusom vykonať privilegovanú operáciu a následné zníženie privilégií, keď sa operácia dokončí, sa označuje ako prechodné zvýšenie privilégií (privilege bracketing), a predstavuje preferovanú metódu používania privilégií aplikáciami. Aby privilégium mohlo byť zvýšené, musí sa nachádzať v sade maximálnych privilégií aplikácie v databáze privilegovaných príkazov. Zvýšením privilégia sa toto privilégium dostane do sady účinných privilégií (EPS) procesu. Zníženie privilégia toto privilégium odstráni zo sady EPS. Nasledujúca ukážka kódu ilustruje prechodné zvýšenie a zníženie privilégií v API rozhraní **auditproc**.

```
priv_raise(PV_AU_ADMIN, -1); /* v prípade potreby zvýšiť privilégium */
auditproc();     /* volať auditovacie systémové volanie */
priv_lower(PV_AU_ADMIN, -1); /* znížiť privilégium */
```

Aplikácie sledujúce RBAC:

V systéme AIX a v systémoch, kde je pre koreňových užívateľov povolené rozšírené RBAC je to tradične tak, že koreňovému alebo koreňom vlastnenému programu **setuid** (s UID=0), ktorý sa nenachádza v databáze privilegovaných príkazov, sú v kerneli priznané všetky privilégiá. Kontroly privilégií v kerneli budú preto vždy úspešné, a to aj v prípade, že požadované privilégium sa nenachádza v sade účinných privilégií (EPS) procesu.

Takéto správanie je naďalej potrebné kvôli podpore existujúcich aplikácií **setuid**, môže však predstavovať bezpečnostné riziko, pretože akýkoľvek program typu **setuid** bude mať všetky práva koreňového užívateľa.

Aby bolo možné prechodné zvýšenie privilégií počas nejakého procesu na systémoch s rozšíreným RBAC a povoleným koreňovým užívateľom, bol zavedený nový bit v štruktúre procesov. Ak je tento bit nastavený, proces sa stane procesom sledujúcim RBAC a účinné UID=0 neposkytne žiadne privilégiá navyše. Tento bit môžete nastaviť v programe systémovým volaním **proc_rbac_op**. Všetky programy **setuid**, ktoré nie sú uvedené v databáze privilegovaných príkazov, môžu použiť túto vlastnosť na zníženie dostupných privilégií a zredukovanie zraniteľnosti. Majte na pamäti, že programy zadefinovaný v databáze privilegovaných príkazov sú automaticky označené ako procesy sledujúce RBAC a budú im priradené iba privilégiá, ktoré sa nachádzajú v databáze.

Nasledujúca ukážka kódu ukazuje, ako môže aplikácia samu seba označiť za aplikáciu sledujúcu RBAC a následne vykonať dočasné zvýšenie privilégií:

```
#include <userpriv.h
#include <sys/priv.h>

privg_t effpriv;

int rbac_flags = SEC_RBACAWARE;

/* Označiť proces ako sledujúci RBAC. */
proc_rbac_op(-1, PROC_RBAC_SET, &rbac_flags);

/* Nastaviť sadu účinných privilégií ako prázdnu. */
priv_clrall(effpriv);
setppriv(-1, &effpriv, NULL, NULL, NULL);

/* Zvýšiť privilégiá, ak je to potrebné. */
priv_raise(PV_AU_ADMIN, -1);
auditproc();

/* Znížiť privilégiá, keď už nie sú potrebné. */
priv_lower(PV_AU_ADMIN, -1);
```

API rozhrania pre RBAC:

Rozhrania API súvisiace s RBAC, ktoré sú dostupné na systéme, sú uvedené v nasledujúcej tabuľke. Viac informácií si vyhľadajte ku konkrétnym rozhraniam API.

API	Popis
checkauths	Porovná odovzdané údaje v zozname autorizácií s autorizáciami priradenými aktuálnemu procesu.
GetUserAuths	Získa sadu autorizácií priradených aktuálnemu procesu.
MatchAllAuths, MatchAllAuthsList, MatchAnyAuths, MatchAnyAuthsList	Porovná autorizácie. Pred týmito API rozhraniami sa uprednostňuje API checkauths.
getauthattr, putauthattr	Dotazuje alebo modifikuje autorizácie definované v databáze autorizácií.
getauthattrs	Získava viaceré atribúty autorizácie z databázy autorizácií.
putauthattrs	Aktualizuje viaceré atribúty autorizácie v databáze autorizácií.
getcmdattr, putcmdattr	Dotazuje alebo modifikuje informácie o zabezpečení príkazu v databáze privilegovaných príkazov.
getcmdattrs	Získava viaceré atribúty príkazov z databázy privilegovaných príkazov.
putcmdattrs	Aktualizuje viaceré atribúty príkazov v databáze privilegovaných príkazov.
getdevattr, putdevattr	Dotazuje alebo modifikuje informácie o zabezpečení zariadenia v databáze privilegovaných zariadení.
getdevattrs	Získava viaceré atribúty zariadení z databázy privilegovaných zariadení.
putdevattrs	Aktualizuje viaceré atribúty zariadení v databáze privilegovaných zariadení.
getpfileattr, putpfileattr	Dotazuje alebo modifikuje informácie o zabezpečení súboru v databáze privilegovaných súborov.
getpfileattrs	Získava viaceré atribúty súborov z databázy privilegovaných súborov.
putpfileattrs	Aktualizuje viaceré atribúty súboru v databáze privilegovaných súborov.
getroleattr, putroleattr	Dotazuje alebo modifikuje roly definované v databáze rolí.
getroleattrs	Získava viaceré atribúty roly z databázy rolí.
putroleattrs	Aktualizuje viaceré atribúty roly v databáze rolí.
getsecorder	Získava poradie domén pre určité bezpečnostné databázy.
setsecorder	Nastavuje poradie domén pre určité bezpečnostné databázy.

Privilégiá v systéme AIX

Privilégiá dostupné v operačnom systéme AIX sú uvedené v nasledujúcej tabuľke. Ku každému privilégiu a s ním súvisiacim systémovým volaniam je uvedený opis. Niektoré privilégiá tvoria hierarchiu, kde jedno privilégium dokáže poskytnúť všetky ostatné práva spojené s iným privilégiom.

Pri kontrole privilégií systém najskôr určí, či má proces najnižšie potrebné privilégium, a potom postupuje v hierarchii vyššie a hľadá silnejšie privilégium. Napríklad proces s privilégiom **PV_AU_** má automaticky privilégiá **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE**, a proces s privilégiom **PV_ROOT** má automaticky všetky privilégiá uvedené nižšie s výnimkou privilégií **PV_SU_**.

Privilégium	Popis	Odkaz na systémové volanie
PV_ROOT	Poskytuje procesu ekvivalent všetkých privilégií uvedených nižšie okrem PV_SU_ (a privilégií, ktorým dominuje)	

Privilégium	Popis	Odkaz na systémové volanie
PV_AU_ADD	Umožňuje procesu zaznamenať/pridať auditovací záznam	auditlog
PV_AU_ADMIN	Umožňuje procesu konfigurovať a dotazovať auditovací systém	audit, auditbin, auditevents, auditobj
PV_AU_PROC	Umožňuje procesu získať alebo nastaviť auditovací stav procesu	auditproc
PV_AU_READ	Umožňuje procesu čítať súbor označený ako auditovací súbor v Dôveryhodný systém AIX	
PV_AU_WRITE	Umožňuje procesu zapísať alebo vymazať súbor označený ako auditovací súbor, alebo označiť súbor ako auditovací súbor v Dôveryhodný systém AIX	
PV_AU_	Ekvivalent ku všetkým vyšším auditovacím privilégiám dohromady (PV_AU_*)	
PV_AZ_ADMIN	Umožňuje procesu modifikovať bezpečnostné tabuľky kernelu	sec_setkst
PV_AZ_READ	Umožňuje procesu obnoviť bezpečnostné tabuľky kernelu	sec_getkat, sec_getkpct, sec_getkpdt, sec_getkrt, atď.
PV_AZ_ROOT	Spôsobuje, že proces počas exec() prejde autorizačnou kontrolou (používa sa na účely dedičnosti)	
PV_AZ_CHECK	Spôsobuje, že proces prejde všetkými autorizačnými kontrolami	sec_checkauth
PV_DAC_R	Umožňuje procesu obísť obmedzenia čítania DAC	access, creat, accessx, open, read, faccessx, mkdir, getea, rename, statx, _sched_getparam, _sched_getscheduler, statea, listea
PV_DAC_W	Umožňuje procesu obísť obmedzenia zápisu DAC	Veľa vyššie uvedených a setea, write, symlink, _setpri, _sched_setparam, _sched_setscheduler, fsetea, rmdir, removeea
PV_DAC_X	Umožňuje procesu obísť obmedzenia vykonania DAC	Veľa vyššie uvedených a execve, symlink, rmdir, chdir, fchdir, ra_execve
PV_DAC_O	Umožňuje procesu obísť obmedzenia vlastníctva DAC	chmod, utimes, setacl, revoke, mprotect
PV_DAC_UID	Umožňuje procesu zmeniť jeho ID užívateľa	setuid, seteuid, setuidx, setreuid, ptrace64
PV_DAC_GID	Umožňuje procesu zmeniť alebo nastaviť nové ID skupiny	setgid, setgidx, setgroups, ptrace64
PV_DAC_RID	Umožňuje procesu zmeniť alebo nastaviť nové ID roly	setroles, getroles
PV_DAC_	Ekvivalent ku všetkým vyšším DAC privilégiám dohromady (PV_DAC_*)	
PV_FS_MOUNT	Umožňuje procesu pripojiť alebo odpojiť súborový systém	vmount, umount
PV_FS_MKNOD	Umožňuje procesu vytvoriť súbor ľubovoľného typu alebo vykonať systémové volanie mknod	mknod

Privilégium	Popis	Odkaz na systémové volanie
PV_FS_CHOWN	Umožňuje procesu zmeniť vlastníctvo súboru	chown, chownx, fchownx, lchown
PV_FS_QUOTA	Umožňuje procesu spravovať operácie súvisiace s kvótami diskového priestoru	quotactl
PV_FS_LINKDIR	Umožňuje procesu vytvoriť pevné prepojenie na adresár	link, unlink, remove
PV_FS_CNTL	Umožňuje procesu vykonať rôzne riadiace operácie na súborovom systéme s výnimkou rozšírenia a zmenšenia	fscntl
PV_FS_RESIZE	Umožňuje procesu vykonať rozširujúce a zmenšujúce typy operácií na súborovom systéme	fscntl
PV_FS_CHROOT	Umožňuje procesu zmeniť jeho koreňový adresár	chroot
PV_FS_PDMODE	Umožňuje procesu vytvoriť alebo nastaviť adresár s oddielmi	pdmkdir
PV_FS_	Ekvivalent ku všetkým vyšším privilégiám súborového systému dohromady (PV_FS_*)	
PV_PROC_PRIV	Umožňuje procesu modifikovať alebo zobraziť sady privilégií asociovaných s procesom	setppriv, getppriv
PV_PROC_PRIO	Umožňuje procesu/vláknu zmeniť prioritu, politiku a iné parametre plánovania	_prio_requeue, _setpri, _setpriority, _getpri, _sched_setparam, _sched_setscheduler, _thread_setsched, thread_boostceiling, thread_setmystate, thread_setstate
PV_PROC_CORE	Umožňuje procesu vytvoriť výpis jadra	gencore
PV_PROC_RAC	Umožňuje procesu vytvoriť viac procesov než koľko stanovuje limit na užívateľa	appsetrlimit, setrlimit64, mlock, mlockall, munlock, munlockall, plock, upfget, upfput, restart, brk, sbrk
PV_PROC_RSET	Umožňuje pripojiť sadu procesov (rset) k procesu alebo vláknu	bindprocessor, ra_attachrset, ra_detachrset, rs_registername, rs_setnameattr, rs_discardname, rs_setpartition, rs_getassociativity, kra_mmapv
PV_PROC_ENV	Umožňuje procesu nastaviť užívateľské informácie v štruktúre užívateľov	ue_proc_register, ue_proc_unregister, usrinfo
PV_PROC_CKPT	Umožňuje procesu vytvoriť bod obnovy alebo reštartovať iný proces	setcrid, restart
PV_PROC_CRED	Umožňuje procesu nastaviť atribúty splnomocnení	__pag_setvalue, __pag_setvalue64, __pag_genpagvalue
PV_PROC_SIG	Umožňuje procesu poslať signál nesúvisiacemu procesu	_sigqueue, kill, signohup, gencore, thread_post, thread_post_many
PV_PROC_TIMER	Umožňuje procesu odovzdať a použiť časovače s jemnou granularitou	appresabs, appresinc, absinterval, incinterval, _poll, _select _timer_settime
PV_PROC_RTCLK	Umožňuje procesu pristupovať k hodinám procesorového času	_clock_getres, _clock_gettime, _clock_settime, _clock_getcpuclockid
PV_PROC_VARS	Umožňuje procesu znovu získať a aktualizovať laditeľné parametre procesu	smttune

Privilégium	Popis	Odkaz na systémové volanie
PV_PROC_PDMODE	Umožňuje procesu zmeniť režim REAL adresára s oddielmi	setppdmode
PV_PROC_	Ekvivalent ku všetkým vyšším procesovým privilégiám dohromady (PV_PROC_*)	
PV_TCB	Umožňuje procesu modifikovať dôveryhodnú cestu ku knižnici kernelu	chpriv, fchpriv
PV_TP	Indikuje, že proces je procesom s dôveryhodnou cestou a umožňuje akcie, ktoré sa obmedzujú na procesy s dôveryhodnou cestou. (Poznámka: Rovnaké ako staré privilégium AIX BYPASS_TPATH)	
PV_WPAR_CKPT	Umožňuje procesu vykonať operáciu vytvorenia bodu obnovy/reštartovania vo WPAR	smcr_proc_info, smcr_exec_info, smcr_mapinfo, smcr_net_oper, smcr_procattr, aio_suspend_io, aio_resume_io
PV_KER_ACCT	Umožňuje procesu vykonať obmedzené operácie týkajúce sa podsystému vedenia účtov	acct, _acctctl, projctl
PV_KER_DR	Umožňuje procesu vyvolať operácie dynamickej rekonfigurácie	_dr_register, _dr_notify, _dr_unregister, dr_reconfig
PV_KER_TIME	Umožňuje procesu modifikovať systémové hodiny a systémový čas	adjtime, appsettimer, _clock_settime
PV_KER_RAC	Umožňuje procesu používať veľké (nestránkovateľné) stránky pre zdieľané pamäťové segmenty	shmctl, vmgetinfo
PV_KER_WLM	Umožňuje procesu inicializovať a modifikovať konfiguráciu WLM	_wlm_set, _wlm_tune, _wlm_assign
PV_KER_EWLM	Umožňuje procesu inicializovať alebo dotazovať prostredie eWLM	
PV_KER_VARS	Umožňuje procesu skúmať alebo nastaviť runtime laditeľné parametre kernelu	sys_parm, getkerninfo, __pag_setname, sysconfig, kunload64
PV_KER_REBOOT	Umožňuje procesu vypnúť systém	reboot
PV_KER_RAS	Umožňuje procesu konfigurovať alebo zapisovať záznamy RAS, protokolovanie chýb, sledovanie, funkcie výpisu pamäte	mtrace_set, mtrace_ctl
PV_KER_LVM	Umožňuje procesu konfigurovať podsystém LVM	
PV_KER_NFS	Umožňuje procesu konfigurovať podsystém NFS	
PV_KER_VMM	Umožňuje procesu modifikovať parametre výmeny a iné laditeľné VMM parametre v kerneli	swapoff, _swapon_ext, vmgetinfo
PV_KER_WPAR	Umožňuje procesu konfigurovať oddiel pracovného zaťaženia	brand, corral_config, corral_delete, corral_modify, wpar_mkdevexport, wpar_rmdevexport, wpar_lsdevexport
PV_KER_CONF	Umožňuje procesu vykonať rôzne operácie konfigurácie systému	sethostname, sethostid, unameu, setdomainname

Privilégium	Popis	Odkaz na systémové volanie
PV_KER_EXTCONF	Umožňuje procesu vykonať rôzne konfiguračné úlohy v rozšíreniach kernelu (pre služby rozšírenia kernelu)	
PV_KER_IPC	Umožňuje procesu zvýšiť hodnotu vyrovnávacej pamäte pre front správ IPC a umožňuje pripojiť shmget s rozsahmi	msgctl, shm_open, shmget, ra_shmget, ra_shmgetv, shmctl
PV_KER_IPC_R	Umožňuje procesu čítať front správ IPC, sadu semaforov alebo segment zdieľanej pamäte	msgctl, __msgrcv, _mq_open, semctl, shmat, shm_open, __semop, shmctl, __semtimedop, sem_post, _sem_wait, __msgrcv, __msgxrcv
PV_KER_IPC_W	Umožňuje procesu zapísať front správ IPC, sadu semaforov alebo segment zdieľanej pamäte	_mq_open, shmat, _sem_open, semctl, shm_open, shmctl, mq_unlink, sem_unlink, shm_unlink, msgctl, __msgsnd
PV_KER_IPC_O	Umožňuje procesu prepísať vlastníctvo DAC na všetkých objektoch IPC	msgctl, semctl, shmctl, fchmod, fchown
PV_KER_SECCONFIG	Umožňuje procesu nastaviť príznaky bezpečnosti kernelu	sec_setsecconf, sec_setrunmode, sec_setsyslab, sec_getsyslab
PV_KER_PATCH	Umožňuje procesu opraviť rozšírenia kernelu	
PV_KER_	Ekvivalent ku všetkým vyšším privilégiám kernelu dohromady (PV_KER_*)	
PV_DEV_CONFIG	Umožňuje procesu konfigurovať kernelové rozšírenia a zariadenia v systéme	sysconfig
PV_DEV_LOAD	Umožňuje procesu zaviesť a uvoľniť kernelové rozšírenia a zariadenia v systéme	sysconfig
PV_DEV_QUERY	Umožňuje procesu dotazovať moduly kernelu	sysconfig
PV_SU_ROOT	Poskytuje procesu všetky privilégiá priradené štandardnému superužívateľovi AIX	
PV_SU_EMUL	Poskytuje procesu všetky privilégiá priradené štandardnému superužívateľovi systému AIX, ak je UID 0	
PV_SU_UID	Spôsobí, že systémové volanie getuid vráti 0	getuidx
PV_SU_	Ekvivalent ku všetkým vyšším privilégiám superužívateľa dohromady (PV_SU_*)	
PV_NET_CNTL	Umožňuje procesu modifikovať sieťové tabuľky	socket, bind, listen, _naccept, econnect, ioctl, rmsock, setsockopt
PV_NET_PORT	Umožňuje procesu naviazať privilegované porty	bind
PV_NET_RAWSOCK	Umožňuje procesu mať priamy prístup k sieťovej vrstve	socket, _send, _sendto, sendmsg, _nsendmsg
PV_NET_CONFIG	Umožňuje procesu konfigurovať sieťové parametre	

Privilégium	Popis	Odkaz na systémové volanie
PV_NET_	Ekvivalent ku všetkým vyšším privilégiám siete dohromady (PV_NET_*)	

Privilégiá uvedené v nasledujúcej tabuľke sú špecifické pre Dôveryhodný systém AIX:

Privilégium Dôveryhodný systém AIX	Popis	Odkaz na systémové volanie
PV_LAB_CL	Umožňuje procesu modifikovať SCL subjektu, podrobené previerke procesu	
PV_LAB_CLTL	Umožňuje procesu modifikovať TCL subjektu, podrobené previerke procesu	
PV_LAB_LEF	Umožňuje procesu čítať súbor kódovania označení	
PV_LAB_SLDG	Umožňuje procesu znížiť úroveň SL, podrobených previerke procesu	
PV_LAB_SLDG_STR	Umožňuje procesu znížiť úroveň SL balíka, podrobeného previerke procesu	
PV_LAB_SL_FILE	Umožňuje procesu zmeniť SL objektu, podrobené previerke procesu	
PV_LAB_SL_PROC	Umožňuje procesu zmeniť SL subjektu, podrobený previerke procesu	
PV_LAB_SL_SELF	Umožňuje procesu zmeniť svoje vlastné SL, podrobené previerke procesu	
PV_LAB_SLUG	Umožňuje procesu zvýšiť úroveň SL, podrobených previerke procesu	
PV_LAB_SLUG_STR	Umožňuje procesu zvýšiť úroveň SL balíka, podrobeného previerke procesu	
PV_LAB_TL	Umožňuje procesu modifikovať TL objektu a subjektu	
PV_LAB_	Ekvivalent ku všetkým vyšším privilégiám označení dohromady (PV_LAB_*)	
PV_MAC_CL	Umožňuje procesu obísť obmedzenia previerky citlivosti	
PV_MAC_R_PROC	Umožňuje procesu obísť obmedzenie čítania MAC pri získavaní informácií o procese, ak platí, že označenie cieľového procesu sa vyskytuje v rámci previerky aktívneho procesu	
PV_MAC_W_PROC	Umožňuje procesu obísť obmedzenie zápisu MAC pri odosielaní signálu procesu, ak platí, že označenie cieľového procesu sa vyskytuje v rámci previerky aktívneho procesu	
PV_MAC_R	Umožňuje procesu obísť obmedzenia čítania MAC	
PV_MAC_R_CL	Umožňuje procesu obísť obmedzenie čítania MAC, keď sa označenie objektu vyskytuje v rámci previerky procesu	

Privilégium Dôveryhodný systém AIX	Popis	Odkaz na systémové volanie
PV_MAC_R_STR	Umožňuje procesu obísť obmedzenie čítania MAC pri čítaní správy zo STREAM, ak platí, že označenie správy sa vyskytuje v rámci previerky procesu	
PV_MAC_W	Umožňuje procesu obísť obmedzenia zápisu MAC	
PV_MAC_W_CL	Umožňuje procesu obísť obmedzenie zápisu MAC, keď sa označenie objektu vyskytuje v rámci previerky procesu	
PV_MAC_W_DN	Umožňuje procesu obísť obmedzenie zápisu MAC, keď označenie procesu dominuje označeniu objektu a označenie objektu sa vyskytuje v rámci previerky procesu	
PV_MAC_W_UP	Umožňuje procesu obísť obmedzenie zápisu MAC, keď označenie objektu dominuje označeniu procesu a označenie objektu sa vyskytuje v rámci previerky procesu	
PV_MAC_OVRRD	Obchádza obmedzenia MAC pre súbory s príznakom, že sú vyňaté spod kontroly MAC	
PV_MAC_	Ekvivalent ku všetkým vyšším privilégiám MAC dohromady (PV_MAC_*)	
PV_MIC	Umožňuje procesu obísť obmedzenia integrity	
PV_MIC_CL	Umožňuje procesu obísť obmedzenia previerky integrity	

Doménové riadenie prístupu RBAC

Riadenie prístupu RBAC (Role-based Access Control), uvedené v AIX 6.1, poskytuje mechanizmus na rozdelenie rozličných úloh nadradeného užívateľa root do rolí, ktoré je možné delegovať na iných užívateľov v systéme. RBAC poskytuje schopnosť delegovať úlohy a zvyšuje bezpečnosť systému, keďže uľahčuje auditovanie a sledovanie aktivít v systéme. RBAC zabezpečuje delegovanie zodpovednosti na iného užívateľa (nazývaného oprávnený užívateľ), neposkytuje však mechanizmus na obmedzenie administračných oprávnení oprávneného užívateľa na konkrétne prostriedky systému. Napríklad, užívateľ, ktorý má administračné oprávnenia pre sieť, môže spravovať všetky sieťové rozhrania v systéme. Nemôžete oprávneného užívateľa obmedziť na konkrétny súbor rozhraní.

Funkcia doménového riadenia prístupu RBAC slúži na vyhradenie prístupu len pre oprávnených užívateľov. Užívatelia a prostriedky systému sa označujú pripojením značiek, ktoré sa nazývajú domény, kým konkrétne prístupové pravidlá určujú prístup k prostriedkom zo strany užívateľov.

Definície

Nasledujúce definície súvisia s prístupovými pravidlami:

subjekt: Subjekt je entita, ktorá vyžaduje prístup k objektu. Príkladom subjektu je proces.

objekt: Objekt je entita, ktorá je držiteľom potrebných informácií. Príkladmi objektov sú súbory, zariadenia a sieťové porty.

doména: Doména je definovaná ako kategória, do ktorej entita patrí. Keď entita patrí do domény, riadenie prístupu k tejto entite sa riadi prístupovými pravidlami nasledujúcim spôsobom:

Prístupové pravidlá

- Subjekt môže pristupovať k objektu, ak má všetky domény, do ktorých objekt patrí. To znamená, že zoznam domén, do ktorých subjekt patrí, musí byť nadmnožinou domén objektu. Toto je štandardné správanie.
- Subjekt môže pristupovať k objektu, ak má s objektom spoločnú aspoň jednu doménu. To znamená, že subjekt a objekt musia patriť aspoň do jednej spoločnej domény. Toto správanie sa závisí od bezpečnostných príznakov objektu.
- Objekt môže zamietnuť prístup k niektorým doménam. Ak objekt definuje množinu domén, ktoré sa nazývajú množiny konfliktov, a ak jedna z domén subjektu je súčasťou množiny konfliktov, objekt môže odmietnuť prístup pre subjekt.

Databáza domén

Domény podporované systémom musia byť uložené v konfiguračnom súbore v adresári /etc/security/domains. Formát odseku v tomto súbore je uvedený nižšie:

```
domain-name:
id = <číslo>
dfltmsg = <správa>
msgcat = <katalóg správ>
msgset = <súbor správ v katalógu>
msgnum = <ID správy v katalógu>
```

S databázou môžete pracovať prostredníctvom príkazov **mkdom** a **chdom**. Na zobrazenie databázy môžete použiť príkaz **lsdom**. Ak chcete vymazať položky z databázy, použite príkaz **rmdom**.

Položky v databáze sa neuplatnia, kým sa nestiahnu do jadra použitím príkazu **setkst**.

V systéme je podporovaných maximálne 1024 domén a najvyššia prípustná hodnota identifikátora domény (atribút ID) je 1024.

Objekty s priradenými doménami

Aby bolo možné priradiť doménu k objektu, tento objekt musí byť definovaný v databáze objektov s priradenými doménami. Domény pre všetky entity v systéme sú uložené v konfiguračnom súbore v adresári /etc/security/domobjs. Formát odseku v tomto súbore je uvedený v príklade nižšie, ktorý ilustruje priradenie domény k objektu.

```
/dev/hrvg:
domains=HR,IT
conflictsets=payroll
objtype=device
secflags=FSF_DOM_ANY
```

domains: Určuje domény, ktoré môžu pristupovať k objektu. Príkladmi domén sú IT, HR a Payroll.

objtype: Určuje typ objektu, ktorý je priradený k doméne. Typy objektov môžu byť: device, file, netint a netport.

conflictsets: Určuje, že ak subjekt patrí do ktorejkoľvek domény uvedenej s týmto atribútom, nemôže pristupovať k objektu.

secflags: Tento príznak určuje špeciálne vlastnosti objektu. Tento príznak sa môže nastaviť na hodnotu **FSF_DOM_ANY** alebo **FSF_DOM_ALL**. Ak je tento príznak nastavený na hodnotu **FSF_DOM_ANY**, subjekt môže pristupovať k objektu, ak obsahuje ktorúkoľvek z domén uvedených s atribútom domains. Ak je tento príznak nastavený na hodnotu **FSF_DOM_ALL**, subjekt musí vyhovieť všetkým doménam v zozname, aby mohol pristupovať k objektu. Ak nezadáte žiadnu hodnotu, použije sa predvolená hodnota **FSF_DOM_ALL**. Atribút **secflag** ovplyvňuje len atribút domains tohto objektu.

Domény je možné priradiť aj k súborom v súborových systémoch. Štandardne musia byť všetky domény objektu podmnožinou domén procesu, aby mohol proces pristupovať k objektu.

1. Zariadenia: Všetky zariadenia (vrátane súborových systémov) je možné priradiť k doménam. Kontroly domén sa vykonávajú pri administračných úlohách, napríklad pri konfigurácii zariadenia.

   ```
   /dev/hrvg:
   domains=HR,IT
   conflictsets=payroll
   objtype=device
   secflags=FSF_DOM_ANY
   ```

2. Sieťové rozhrania: Keď sa sieťové rozhrania (napríklad en0) priradia k doméne, administračné úlohy, ako napríklad vypnutie rozhrania, budú vyžadovať vykonanie kontrol domén na rozhraní.

   ```
   en0:
   domains=NETIF,ADMIN
   objtype=netint
   flags=FSF_DOM_ALL
   ```

3. Sieťové porty: K doméne môžete priradiť porty TCP a UDP. Keď sa aplikácia pokúsi pripojiť k portu, vykonajú sa kontroly domén.

   ```
   TCP_<číslo_portu>:
   domains=NETIF,ADMIN
   type=netport
   flags=FSF_DOM_ALL
   ```

4. Procesy: Proces dedí domény užívateľa, v ktorého mene sa tento proces vykonáva. Keď sa užívateľ prihlási, domény užívateľa vlastní proces rozhrania shell užívateľa. Keď sa nastavia domény, tieto domény procesu sa budú uplatňovať počas celej svojej životnosti. Domény procesu nie je možné zmeniť prostredníctvom žiadneho užívateľského rozhrania alebo systémového volania. Jediný proces, ktorý môže nastaviť domény, je proces prihlásenia. Procesy nemajú atribúty **conflictset** a **secflags**.

Aktuálne obmedzenia

Doménové riadenie prístupu RBAC má v súčasnosti nasledujúce obmedzenia:

- Konfiguračné súbory domény sú v súčasnosti podporované len v lokálnom systéme, nie však na serveri LDAP (Lightweight Directory Access Protocol).
- Domény RBAC nie sú podporované v rámci oddielov pracovného zaťaženia AIX (WPAR).
- Domény RBAC sa nemôžu vzťahovať na prechodné súbory.

Požiadavka Enhanced RBAC

Doménové riadenie prístupu RBAC sa vytvára v Enhanced RBAC a vyžaduje povolenie funkcie Enhanced RBAC v systéme, aby sa uplatnilo.

Bezpečnostné tabuľky kernelu (KST)

Domény a objekty s priradenými doménami, ako sú definované v databáze domén a databáze objektov domén, sa uplatnia po stiahnutí do jadra použitím príkazu **setkst**. Tieto dve tabuľky sa nazývajú Kernel Domain Table (KDOMT) and Kernel Domain Object Table (KDOT).

Ďalšie informácie o bezpečnostných tabuľkách jadra a príkaze **setkst** nájdete v téme Role Based Access Control (RBAC) v publikácii AIX Security Guide.

Príkazy pre domény

Nasledujúca tabuľka uvádza príkazy pre domény RBAC, ktoré sa poskytujú s operačným systémom AIX na správu a používania rámca doménového riadenia prístupu RBAC:

Príkaz	Popis
mkdom	Vytvorí novú doménu
lsdom	Zobrazí atribúty domény
rmdom	Odstráni doménu
chdom	Zmení atribúty domény
setsecattr	Nastaví bezpečnostné atribúty databázy objektov domény
lssecattr	Zobrazí bezpečnostné atribúty databázy objektov domény
rmsecattr	Odstráni definíciu databázy objektov domény
setkst	Odošle položky v databázach domén RBAC na úrovni užívateľa do bezpečnostných tabuliek jadra

Doménové súbory pre RBAC

Nasledujúca tabuľka uvádza súbory pre RBAC, ktoré sa poskytujú s operačným systémom AIX na konfiguráciu a ukladanie databázových informácií:

Súbor	Popis
/etc/security/domains	Databáza domén
/etc/security/domobjs	Databáza objektov domény

Používanie domén

Definovanie domén: Domény sa definujú v databáze domén s použitím príkazu **mkdom**.

```
mkdom id=24 HR
```

Priradenie domén: Domény môžete priradiť k rôznym entitám, ako sú užívatelia, súbory, zariadenia, sieťové porty a rozhrania. Všetky entity okrem užívateľov podporujú množiny konfliktov a bezpečnostné príznaky (**secflags**).

Užívatelia: Užívateľov je možné priradiť k doménam použitím príkazov **chuser** a **chsec**.

Syntax:

```
chuser domains = <zoznam domén oddelených čiarkou> meno užívateľa
```

Príklad:

```
chuser domains=INET john
```

Počas prihlasovania sa aktivujú domény priradené k užívateľovi. Ak sa domény zmenili, kým bola vaša relácia aktívna, musíte sa prihlásiť znova, aby sa uplatnili nové domény.

Objekty: Aby bolo možné obmedziť prístup k objektom prostredníctvom domén, tieto objekty sa musia definovať v databáze objektov domény použitím príkazu **setsecattr**.

Syntax:

```
setsecattr -o domains=<zoznam povolených domén oddelených čiarkou>
conflictsets=<zoznam obmedzených domén oddelených čiarkou>
secflags=<FSF_DOM_ALL alebo FSF_DOM_ANY>
objtype=<file alebo device alebo netint alebo netport>
object-path
```

Príklad:

```
setsecattr -o domains=INET,WEB conflictsets=DB secflags=FSF_DOM_ANY objtype=netint en0
```

Zoznamy riadenia prístupov (ACL)

ACL zvyčajne tvorí séria položiek, ktoré sa nazývajú položkami riadenia prístupu (ACE). Každá položka ACE definuje prístupové práva pre užívateľa vo vzťahu k objektu.

Keď dôjde k pokusu o prístup, operačný systém použije položku ACE asociovanú s týmto objektom a zistí, či dotyčný používateľ má na prístup práva. Tieto zoznamy ACL a súvisiace kontroly prístupu tvoria jadro mechanizmu Discretionary Access Control (DAC) podporovaného operačným systémom AIX.

Operačný systém podporuje niekoľko typov systémových objektov, ktoré umožňujú užívateľským procesom ukladať alebo presúvať informácie. Najdôležitejšími typmi objektov s riadeným prístupom sú nasledovné:

- Súbory a adresáre
- Pomenované dátovody
- Objekty medziprocesovej komunikácie (IPC) ako napríklad fronty správ, segmenty zdieľanej pamäte a semafory

Kontrola prístupových práv pre tieto objekty sa vykonáva na úrovni systémových volaní pri prvom prístupe k objektu. Pretože k objektom SVIPC sa pristupuje nezávisle, kontrola sa vykonáva pre každý prístup. V prípade objektov s názvami súborového systému je potrebné, aby bolo možné rozlíšiť názov aktuálneho objektu. Názvy sa rozlišujú buď relatívne (voči pracovnému adresáru procesu), alebo absolútne (voči koreňovému adresáru procesu). Celé rozlišovanie názvu začína vyhľadaním jedného z týchto adresárov.

Mechanizmus riadenia prístupu na základe uváženia užívateľa umožňuje efektívne ovládanie prístupu k informačným prostriedkom a poskytuje osobitnú ochranu dôvernosti a integrity informácií. Mechanizmy vlastníkom riadeného prístupu sú však len tak efektívne, ako ich užívatelia navrhnú. Všetci užívatelia by mali chápať princípy prideľovania a zakazovania prístupových práv a ich nastavovania.

Napríklad zoznam ACL asociovaný s objektom súborového systému (súbor alebo adresár) by mohol, pokiaľ ide o prístup k danému objektu, vynucovať prístupové práva pre rozličných používateľov. Je možné, že takýto ACL by mohol vynútiť pre rôznych užívateľov rôzne úrovne prístupových práv, napríklad na čítanie alebo zápis.

Najčastejšie bude mať každý objekt zadefinovaného vlastníka a v niektorých prípadoch bude asociovaný s primárnou skupinou. Vlastník konkrétneho objektu môže riadiť jeho voliteľné prístupové atribúty. Atribúty vlastníka sú nastavené na vytváranie efektívneho ID užívateľa procesu.

Nasledujúci zoznam obsahuje atribúty riadenia priameho prístupu pre rozličné typy objektov:

Vlastník
: Tvorca alebo majiteľ môže zmeniť vlastníctvo objektov SVIPC (System V Interprocess Communication). Objekty SVIPC majú priradeného tvorcu, ktorý má všetky práva vlastníka (vrátane autorizácie prístupu). Totožnosť autora zmeniť nemožno dokonca ani s oprávnením úrovne root.

  Objekty SVIPC sú inicializované na efektívne ID skupiny vytvárajúceho procesu. Pre systémové objekty súboru sú atribúty riadenia priameho prístupu inicializované do efektívneho skupinového ID procesu vytvárania alebo skupinového ID rodičovského adresára (čo je určené skupinovým dedičným príznakom rodičovského adresára).

Skupina
: Vlastník objektu môže zmeniť skupinu. Takáto nová skupina musí byť buď platným skupinovým ID procesu vytvárania alebo skupinovým ID rodičovského adresára. (Ako už bolo uvedené, objekty SVIPC majú priradenú skupinu vytvárania, ktorú nemožno meniť a zdieľajú autorizáciu prístupu skupiny objektov.)

Režim
: Príkaz **chmod** (v numerickom režime v osmičkovom zápise) môže nastaviť základné povolenia a atribúty. Subrutina **chmod**, ktorú volá daný príkaz, zakazuje rozšírené povolenia. Ak používate numerický režim príkazu **chmod** v súbore, ktorý má ACL, rozšírené oprávnenia budú vypnuté. Symbolický režim príkazu **chmod** vypína rozšírené zoznamy ACL typuNSF4, ale nevypína rozšírené oprávnenia pre zoznamy ACL typu AIXC. Informácie o numerickom a symbolickom režime nájdete v **chmod**.

Mnohé objekty v operačnom systéme, napríklad sokety a objekty súborového systému, majú na rôzne účely asociované zoznamy ACL. Podrobnosti o zoznamoch ACL pre tieto typy objektov sa môžu od seba vzájomne líšiť.

Operačný systém AIX tradične podporoval bity režimu na riadenie prístupu do objektov systému súborov. Podporuje tiež jedinečnú formu ACL vo všetkých bitoch režimu. Tento ACL pozostával zo základných bitov režimu a rátal tiež s definovaním viacerých položiek ACE, pričom každá položka ACE definovala prístupové práva pre používateľa alebo skupinu vo všetkých bitoch režimu. Tento klasický typ správania sa zoznamu ACL bude naďalej podporovaný pod názvom AIXC ACL.

Podotýkame, že podpora zoznamu riadenia prístupu na objektoch súborového systému závisí od základného fyzického súborového systému (PFS). PFS musí rozpoznávať údaje ACL a musí ich vedieť uložiť, načítať a vynútiť prístupy pre rozličných používateľov. Je možné, že niektoré fyzické súborové systémy nepodporujú vôbec nijaké zoznamy ACL (môžu podporovať základné bity režimu) - na rozdiel od fyzických súborových systémov, ktoré podporovali viacero typov ACL. Niektoré súborové systémy v systéme AIX boli vylepšené tak, aby podporovali rôzne typy zoznamov ACL. JFS2 a GPFS budú schopné podporovať aj typ ACL založený na protokole NFS verzie 4. Tento ACL bol v AIX nazvaný typom NFS4 ACL. Tento typ ACL sa z väčšej časti pridržiava špecifikácií definície ACL v protokole NFS version 4. V porovnaní s typom AIXC ACL podporuje aj jemnejšie riadenie prístupu a poskytuje také schopnosti ako napríklad dedičnosť.

Podpora rámca viacerých typov zoznamov riadenia prístupov

Počnúc verziou 5.3.0, operačný systém AIX podporuje infraštruktúru pre rôzne typy zoznamov riadenia prístupu (ACL) pre rôzne objekty súborového systému v rámci operačného systému.

Táto infraštruktúra zabezpečuje jednotné metódy riadenia ACL bez ohľadu na typ ACL, ktorý je s objektom asociovaný. Rámec obsahuje tieto komponenty:

Administratívne príkazy ACL
: Sú to napríklad príkazy **aclget**, **aclput**, **acledit**, **aclconvert**, **aclgetttypes**. Tieto príkazy volajú rozhrania knižnice, ktoré vyvolajú moduly špecifické pre typy ACL.

Rozhrania knižnice ACL
: Rozhrania knižnice ACL konajú ako front-endy aplikácií, ktoré potrebujú prístup do ACL.

Dynamicky zavádzateľné moduly ACL špecifické pre daný typ ACL
: Operačný systém AIX poskytuje niekoľko modulov špecifických pre typ ACL pre klasické ACL systému AIX (**AIXC**) a ACL NFS4 (**nfs4**).

Binárna kompatibilita:

Neexistujú žiadne problémy s kompatibilitou aplikácií, ktoré sa spúšťajú z existujúcich súborových systémov JFS2, bez ohľadu na to, či existujú zoznamy riadenia prístupu systému AIX.

Podotýkame však, že aplikácie prípadne môžu naraziť na problém s prístupom k súborom, a to v prípade, ak sa stretnú s objektmi súborového systému, ktoré majú asociované oveľa prísnejšie ACL (ako napríklad NFS4). Jednoduchou kontrolou zistite, či bude existujúci súbor vyžadovať v zozname riadenia prístupu NFS4 povolenie na čítanie.

Typy podporovaných zoznamov riadenia prístupu (ACL) v operačnom systéme AIX

Operačný systém AIX v súčasnosti podporuje typy zoznamov riadenia prístupu AIXC a NFS4.

Ak sme už povedali, podporuje aj infraštruktúru pre pridávanie akéhokoľvek ďalšieho typu ACL, ktorý je podporovaný základným fyzickým súborovým systémom. Podotýkame, že súborový systém FS2 PFS implicitne podporuje zoznamy NFS4 ACL, ak je inštancia súborového systému vytvorená pomocou Extended Attributes Version 2.

Zoznam riadenia prístupov AIXC:

Typ zoznamu riadenia prístupu AIXC predstavuje typ ACL podporovaný vo vydaniach systému AIX starších ako 5.3.0. Zoznamy riadenia prístupov AIXC zahŕňajú základné a rozšírené povolenia.

Typ zoznamu riadenia prístupu (ACL) AIXC predstavuje typ ACL podporovaný vo vydaniach systému AIX starších ako 5.3.0. AIXC ACL zahŕňajú základné a rozšírené povolenia. Systém súborov JFS2 povoľuje pre zoznamy riadenia prístupov (ACL) AIXC maximálnu veľkosť 4 KB.

Nastavenie základných povolení pre zoznamy riadenia prístupov AIXC

Základné oprávnenia predstavujú tradičné režimy prístupu k súborom priradené vlastníkovi súboru, skupine súboru a ďalším užívateľom. Režimy prístupu sú nasledovné: čítanie (r), zápis (w) a spustenie/prehľadávanie (x).

V ACL sú základné oprávnenia v nasledovnom formáte, pričom parameter *Režim* je vyjadrený ako rwx (pomlčka (-) nahrádza každé nešpecifikované oprávnenie:

```
base permissions:
   owner(name): Mode
   group(group): Mode
   others: Mode
```

Nastavenie atribútov zoznamu riadenia prístupov AIXC

Do AIXC ACL možno pridať tieto atribúty:

setuid (SUID)
Bit režimu Set-user-ID. Tento atribút nastavuje ID efektívnych a uložených užívateľov procesu do ID vlastníka súboru v čase spustenia.

setgid (SGID)
Bit režimu Set-group-ID. Tento atribút nastavuje ID efektívnych a uložených skupín procesu do ID skupiny súboru v čase spustenia.

savetext (SVTX)
V prípade adresárov indikuje, že súbory v danom adresári môžu pripájať a odpájať len vlastníci súborov.

Tieto atribúty sa pridávajú v nasledovnom formáte:

```
attributes: SUID, SGID, SVTX
```

Nastavenie rozšírených povolení pre zoznam riadenia prístupov AIXC

Rozšírené oprávnenia umožňujú vlastníkovi súboru presnejšie definovať prístup k tomuto súboru. Rozšírené oprávnenia povolením, zakázaním alebo zadaním režimov prístupu pre konkrétne osoby, skupiny alebo kombinácie osôb a skupín upravujú základné oprávnenia pre prístup k súborom (owner, group, others). Oprávnenia sa modifikujú prostredníctvom kľúčových slov.

Kľúčové slová **permit**, **deny** a **specify** sú definované nasledovne:

permit Povolí užívateľovi alebo skupine zadaný prístup k súboru

deny Zamedzí užívateľovi alebo skupine použitie zadaného prístupu k súboru

specify Presne definuje prístup k súboru pre užívateľa alebo skupinu

Ak je užívateľovi zamedzené v konkrétnom prístupe prostredníctvom kľúčového slova **deny** alebo **specify**, žiadna ďalšia zadaná hodnota nemôže tento zákaz obísť.

Aby rozšírené oprávnenia nadobudli účinnosť, musí byť v zozname prístupových práv zadané kľúčové slovo **enabled**. Predvolenou hodnotou je kľúčové slovo **disabled**.

V zozname prístupových práv sa rozšírené oprávnenia uvádzajú v nasledovnom formáte:

```
extended permissions:
  enabled | disabled
    permit   Mode  UserInfo...
    deny     Mode  UserInfo...
    specify  Mode  UserInfo...
```

Pre každé zadané kľúčové slovo **permit**, **deny** alebo **specify** použite samostatný riadok. Parameter *Mode* je vyjadrený ako **rwx** (s pomlčkou (-) nahrádzajúcou každé nešpecifikované povolenie). Parameter *UserInfo* je vyjadrený ako u:UserName alebo g:GroupName, alebo čiarkou oddelená kombinácia u:UserName a g:GroupName.

Poznámka: Pretože proces má iba jedno ID používateľa, ak sa v položke objaví viac ako jedno meno používateľa, nemôže byť takáto položka použitá pri rozhodnutí o riadení prístupu.

Textové zastúpenie zoznamu riadenia prístupov AIXC

Nasledujúca stanza zobrazuje textové zastúpenie zoznamu riadenia prístupov AIXC:

```
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...
```

Binárny formát zoznamu riadenia prístupov AIXC

Binárny formát zoznamu riadenia prístupu AIXC je definovaný v súbore /usr/include/sys/acl.h a je implementovaný v aktuálnom vydaní operačného systému AIX.

Príklad zoznamu riadenia prístupov AIXC

Nasleduje príklad zoznamu ACL AIXC:

```
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:peter, g:gateway, g:mail
        permit  rw-  g:account, g:finance
```

Položky ACL sú popísané nasledovne:

- Prvý riadok indikuje, že je zapnutý bit **setuid**.
- Ďalší riadok zavádzajúci základné povolenia je voliteľný.
- Ďalšie tri riadky určujú základné oprávnenia. Meno vlastníka a názov skupiny v zátvorkách sú uvedené len pre informačné účely. Zmena týchto názvov nemá za následok zmenu vlastníka alebo skupiny súboru. Tieto atribúty súboru môžu zmeniť len príkazy **chown** a **chgrp**.
- Ďalší riadok zavádzajúci rozšírené povolenia je voliteľný.
- Ďalší riadok indikuje, že nasledujúce rozšírené oprávnenia sú povolené.
- Posledné štyri riadky predstavujú položky rozšírených oprávnení. Prvá položka rozšírených oprávnení prideľuje užívateľovi *dhs* oprávnenie na čítanie (r) a zápis (w) do súboru.

- Druhá položka rozšírených oprávnení zamedzuje užívateľovi *chas* prístup na čítanie (r), avšak len v prípade, že je členom skupiny *system*.
- Tretia položka rozšírených oprávnení určuje, že pokiaľ užívateľ *peter* je členom skupiny *gateway* aj skupiny *mail*, bude mať prístup na čítanie (r). Ak užívateľ *john* nie je členom oboch skupín, toto rozšírené povolenie sa nepoužije.
- Posledná rozšírená položka udeľuje ľubovoľnému užívateľovi v skupinách *account* a *finance* povolenie na čítanie (r) a zápis (w).

  Poznámka: Na proces požadujúci prístup k riadenému objektu možno uplatniť viac rozšírených položiek s obmedzujúcimi položkami, ktoré budú mať prednosť pred povoľujúcimi režimami.

Kompletnú syntax nájdete pri príkaze **acledit** v časti *Commands Reference*.

Zoznam riadenia prístupov NFS4:

Operačný systém AIX podporuje aj typ zoznam riadenia prístupu (ACL) NFS4.

Typ ACL NFS4 implementuje ovládanie riadenia podľa špecifikácie v *NFS (Network File System) Verzia 4 protokolu RFC 3530.* . Systém súborov JFS2 povoľuje pre ACL NFS4 maximálnu veľkosť 64KB.

ACL NFS V4 sú podporované len klientom NFS V4. Cachefs ani Proxy nepodporujú ACL NFS V4.

Textové zastúpenie ACL NFS4

Textový ACL NFS V4 predstavuje zoznam ACE (položiek riadenia prístupu) každý ACE na jeden riadok. ACE má štyri prvky v tomto formáte:

```
IDENTITY    ACE_TYPE      ACE_MASK     ACE_FLAGS

kde:
IDENTITY => má formát 'IDENTITY_type:(IDENTITY_name alebo IDENTITY_ID alebo IDENTITY_who):'
         kde:
         IDENTITY_type => Jeden z týchto typov identity:
                 u : user
                 g : group
                 s : special who string (IDENTITY_who must be a special who)
                         IDENTITY_name => názov užívateľa/skupiny
                         IDENTITY_ID   => ID užívateľa/skupiny
                         IDENTITY_who  => špeciálny reťazec who (napríklad OWNER@, GROUP@, EVERYONE@)
ACE_TYPE => jeden z týchto typov ACE:
                a : allow
                d : deny
                l : alarm
                u : audit
ACE MASK => Jeden alebo viacero nasledujúcich kľúčov hodnoty masky bez oddeľovača:
                r : READ_DATA   or LIST_DIRECTORY
                w : WRITE_DATA      or ADD_FILE
                p : APPEND_DATA     or ADD_SUBDIRECTORY
                R : READ_NAMED_ATTRS
                W : WRITE_NAMED_ATTRS
                x : EXECUTE         or 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 (Optional) => Jeden alebo viacero nasledujúcich kľúčov atribútu bez oddeľovača:
                fi : FILE_INHERIT
                di : DIRECTORY_INHERIT
                oi : INHERIT_ONLY
                ni : NO_PROPAGATE_INHERIT
                sf : SUCCESSFUL_ACCESS_ACE_FLAG
                ff : FAILED_ACCESS_ACE_FLAG
```

Poznámka: Pokiaľ ide o kľúč hodnoty SYNCHRONIZE Ace_Mask, s, AIX nevykoná žiadnu akciu v súvislosti s týmto kľúčom hodnoty. Operačný systém AIX ukladá a uchováva kľúč hodnoty s, ale tento kľúč hodnoty nemá žiadny význam pre operačný systém AIX.

Keď je WRITE_OWNER Ace_Mask nastavené na Ace_Type allow, užívatelia môžu meniť vlastníctvo súboru len pre seba.

Vymazanie súboru závisí od dvoch položiek ACE, a to položky DELETE objektu, ktorý má byť vymazaný, a položky DELETE_CHILD jeho rodičovského adresára. Operačný systém AIX poskytuje užívateľom dva režimy prevádzky. V *bezpečnom* (secure) režime sa položka DELETE správa podobne ako AIXC ACL. V režime *kompatibility* sa DELETE správa rovnako ako v ostatných hlavných implementáciách NFS4 ACL. Kompatibilný režim zapnete nasledovne pomocou príkazu **chdev**:

```
chdev -l sys0 -a nfs4_acl_compat=compatible
```

Po spustení príkazu **chdev** musíte systém opätovne zaviesť, aby zmena konfigurácie nadobudla platnosť.

Ak zmeníte režim, musíte brať ohľad na to, že zoznamy riadenia prístupu NFS4 vygenerované operačným systémom AIX v bezpečnom režime nemusia byť akceptované inými platformami ani vtedy, ak bol systém znova uvedený do režimu kompatibility.

```
Príklad:
        u:user1(aa@ibm.com):     a    rwp    fidi
        *s:(OWNER@):             d    x      dini          * This line is a comment
        g:staff(jj@jj.com):      a    rx
        s:(GROUP@):              a    rwpx   fioi
        u:2:                     d    r     di             * This line shows user bin (uid=2)
        g:7:                     a    ac    fi             * This line shows group security (gid=7)
        s:(EVERYONE@):           a    rca   ni
```

Binárny formát pre ACL NFS4

Binárny formát zoznamu riadenia prístupu NFS4 je definovaný v súbore /usr/include/sys/acl.h a je implementovaný v aktuálnom vydaní operačného systému AIX.

Príklad ACL NFS4

Nasledujúci príklad zobrazuje NFS4 použitý na adresári (napríklad j2eav2/d0):

```
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
```

Položky ACL sú opísané takto:

- Prvá položka ACE uvádza, že vlastník má nasledujúce privilégiá na /j2eav2/d0 a použijú sa všetci potomkovia vytvorení po tomto ACL:
  - 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
- Druhá položka ACE určuje, že majiteľovi bolo zamietnuté privilégium pre DELETE_CHILD (vymazanie súborov alebo podadresárov vytvorených pod /j2eav2), ale vlastník ich ešte stále môže vymazať vzhľadom na prvú položku ACE, ktorá povoľuje vlastníkovi privilégium pre DELETE_CHILD.
- Tretia položka ACE určuje, že všetkým členom skupiny pre objekt (/j2eav2/d0) je odmietnuté privilégium pre EXECUTE (=SEARCH_DIRECTORY), ale vlastníkovi je toto privilégium stále povolené prvou položkou ACE. Táto ACE nemôže byť propagovaná pre všetkých svojich potomkov, pretože je zadaný príznak NO_PROPAGATE_INHERIT. Táto ACE sa použije len pre adresár /j2eav2/d0 a jej súbory okamžitých potomkov a podadresáre.
- Štvrtá položka ACE určuje, že každý člen skupiny objektov (/j2eav2/d0) má povolené privilégium pre READ_DATA (= LIST_DIRECTORY) a EXECUTE (=SEARCH_DIRECTORY) na /j2eav2/d0 a všetkých svojich potomkoch. Avšak, kvôli tretej položke ACE nemajú členovia skupiny (s výnimkou vlastníka) povolené privilégium pre EXECUTE (=SEARCH_DIRECTORY) na adresári /j2eav2/d0 a súboroch okamžitých potomkov a podadresároch.
- Piata položka ACE určuje, že každý má povolené privilégium pre adresár READ_ACL na /j2eav2/d0 a ľubovoľných potomkoch vytvorených po použití tohto zoznamu ACL.
- Šiesta položka ACE určuje, že každému bude zamietnuté privilégium pre WRITE_ACL na adresári /j2eav2/d0 a ľubovoľnom potomkovi. Vlastník má vždy privilégium pre WRITE_ACL na súboroch a adresároch s NFS4 ACL.
- Siedma položka ACE určuje, že užívateľ1 má privilégium pre WRITE_DATA (=ADD_FILE) a APPEND_DATA (= ADD_SUBDIRECTORY) na všetkých potomkoch adresára /j2eav2/d0, ale nie na samotný adresár /j2eav2/d0.
- Deviata položka ACE uvádza, že všetci členovia skupiny1 majú odmietnuté privilégium pre WRITE_DATA (=ADD_FILE) a APPEND_DATA (= ADD_SUBDIRECTORY). Táto položka ACE sa nepoužíva pre majiteľa, aj keby patril k skupine1 vzhľadom na prvú položku ACE.
- Deviata položka ACE určuje, že užívateľ s **UID 101** má privilégium pre WRITE_ACL, ale nikto s výnimkou vlastníka nemá privilégium pre WRITE_ACL kvôli šiestej položke ACE.
- Desiata položka ACE určuje, že všetci členovia skupiny s **GID 100** sú odmietnutí pre READ_ACL, ale budú mať toto privilégium kvôli piatej položke ACE.

Manažment zoznamu riadenia prístupu

Zoznamy ACL môžete zobraziť a nastaviť pomocou rôznych príkazov.

Programátori aplikácií a ďalší vývojári podsystémov môžu používať rozhrania knižnice ACL a rutiny konverzie ACL opísané v tejto časti.

Príkazy administrácie ACL

Nasledujúce príkazy môžete použiť na prácu s ACL pre objekt systému súborov:

aclget Zapisuje do štandardného výstupu ACL objektu súboru s názvom *FileObject* prezentovaný v čitateľnom formáte alebo zapisuje to isté do výstupného súboru s názvom *outAclFile*.

aclput Nastavuje ACL *FileObject* na systéme súborov pomocou vstupu zadaného prostredníctvom štandardného vstupu alebo *inAclFile*.

acledit Otvára editor na úpravu ACL zadaného *FileObject*.

aclconvert
Konvertuje ACL z jedného typu do iného. Ak nie je konverzia podporovaná, tento príkaz bude neúspešný.

aclgettypes
Získava typy ACL podporované cestou systému súborov.

Rozhrania knižnice ACL

Rozhrania knižnice ACL pracujú ako front-endy aplikácií, ktoré potrebujú prístup do zoznamov ACL. Aplikácie (vrátane vyššie uvedených generických príkazov správy ACL) nevyvolávajú priamo nezdokumentované systémové volania ACL; namiesto toho vstupujú do generických systémových volaní a zavádzateľných modulov špecifických pre typ cez rozhrania knižnice. Toto ochráni programátorov aplikácií zákazníka pred komplikáciami pri používaní zavádzateľných modulov a minimalizuje problémy so spätnou binárnou kompatibilitou v prípade budúcich vydaní systému AIX.

Nasledujúce rozhrania knižnice volajú systémové volania.

aclx_fget a aclx_get
: Funkcie **aclx_get** a **aclx_fget** načítajú informácie o ovládaní prístupu pre objekt systému súborov a umiestnia ich do pamäťového regiónu zadaného pomocou **acl**. Informácie o veľkosti a type pre **acl** sú uložené v ***acl_sz** a ***acl_type**.

aclx_fput a aclx_put
: Funkcie **aclx_put** a **aclx_fput** ukladajú informácie o ovládaní prístupu zadané v **acl** pre objekt vstupného súboru. Tieto funkcie nevykonávajú konverzie typu ACL; aby bola vykonaná konverzia typu ACL, volajúci musí explicitne zavolať funkciu **aclx_convert**.

aclx_gettypes
: Funkcia **aclx_gettypes** získava zoznam typov ACL podporovaných na určitom systéme súborov. Typ systému súborov môže simultánne podporovať viac než jeden typ ACL. Každý objekt systému súborov je spojený s jedinečným typom ACL patriacim do zoznamu typov ACL podporovaných systémom súborov.

aclx_gettypeinfo
: Funkcia **aclx_gettypeinfo** získava charakteristiky a schopnosti typu ACL na systéme súborov zadanom cestou. Všimnite si, že charakteristiky ACL budú mať normálne typ štruktúry údajov, čo je špecifické pre každý konkrétny typ ACL. Štruktúry údajov používané pre AIXC a NFS4 ACL budú opísané v samostatnom dokumente.

aclx_print a aclx_printStr
: Tieto dve funkcie konvertujú ACL daný v binárnom formáte do textovej reprezentácie . a sú volané príkazmi **aclget** a **acledit**.

aclx_scan a aclx_scanStr
: Tieto dve funkcie konvertujú danú textovú reprezentáciu ACL do binárneho formátu.

aclx_convert
: Konvertuje ACL z jedného typu na iný. Táto funkcia sa používa na implicitnú konverziu pomocou príkazov, napríklad **cp**, **mv** alebo **tar**.

Konverzia ACL

Konverzia ACL umožňuje konvertovať jeden typ ACL na iný. Podpora viacerých typov ACL závisí od toho, ktoré typy ACL sú podporované na konkrétnom fyzickom systéme súborov. Nie všetky typy ACL sú podporované nie všetkými systémami súborov. Napríklad systém súborov jedna by mohol podporovať len typy AIXC ACL a systém súborov dva môže podporovať typy AIXC a NFS4 ACL. Môžete skopírovať AIXC ACL medzi týmito dvomi systémami súborov, ale ak chcete kopírovať NFS ACL zo systému súborov dva do systému súborov jedna, musíte používať konverziu ACL. Konverzia ACL čo najviac chráni informácie o ovládaní prístupu.

Poznámka: Proces konverzie je približný a mohol by viesť k strate informácií o ovládaní prístupu, čo je potrebné zvážiť pri plánovaní konverzií vášho ACL.

Konverzia ACL v operačnom systéme AIX je podporovaná s nasledujúcou infraštruktúrou:

Knižničné rutiny
: Tieto rutiny a rámec ACL užívateľskej úrovne umožňujú konverziu ACL z jedného typu ACL na iný.

Príkaz aclconvert
Tento príkaz konvertuje ACL.

Príkazy aclput a acledit
Tieto príkazy sa používajú na modifikáciu typov ACL.

Príkazy cp a mv
Tieto príkazy boli povolené pre spracúvanie viacerých typov ACL a vykonávanie akejkoľvek internej konverzie ACL podľa potreby.

Príkaz backup
Tento príkaz v prípade požiadavky na zálohovanie vo formáte legacy konvertuje informácie o ACL do známeho typu a formy (typ AIXC ACL). Ak chcete načítať ACL v jeho natívnom formáte, zadajte voľbu **-U**. Bližšie informácie nájdete v časti o zálohovaní.

Každý typ ACL je jedinečný a prepracovanosť masiek ovládania prístupov sa medzi jednotlivými typmi ACL značne líši. Algoritmy konverzie sú približné a nie sú ekvivalentom manuálnej konverzie ACL. V niektorých prípadoch nebude konverzia presná. Napríklad NFS4 ACL nemožno skutočne konvertovať na AIXC ACL, pretože NFS4 ACL poskytujú až 16 prístupových masiek a majú dedičské vlastnosti, ktoré nie sú podporované v type AIXC ACL). Ak sa obávate straty informácií o ovládaní prístupov, nemali by ste používať rozhrania a zariadenia na konverziu ACL.

Poznámka: Algoritmy konverzie ACL majú vlastnícku povahu a podliehajú zmene.

S bity a zoznamy prístupových práv
Môžete použiť programy **setuid** a **setgid** a použiť S bity pre zoznamy ACL.

Používanie programov setuid a setgid

Mechanizmus povoľovacích bitov umožňuje vo väčšine prípadov efektívne ovládanie prístupu k prostriedkom. Pre presnejšie ovládanie prístupu poskytuje operačný systém programy **setuid** a **setgid**.

Operačný systém AIX definuje identitu iba v zmysle identifikátorov UID a GID. Typy ACL, ktoré nedefinujú identity prostredníctvom identifikátorov UID a GID, sa mapujú na model identity systému AIX. Napríklad typ NFS4 ACL definuje užívateľskú identitu ako reťazce v tvare user@domain a tento reťazec je mapovaný do numerických UID a GID.

Väčšina programov beží s užívateľskými alebo skupinovými prístupovými právami užívateľa, ktorý ich vyvolal. Vlastníci programov môžu priradiť prístupové práva užívateľa, ktorý ich spustil tak, že program nastavia ako **setuid** alebo **setgid**. To znamená, že program bude mať vo svojom poli oprávnení nastavený bit setuid alebo setgid. Keď je daný program spustený týmto procesom, proces získa prístupové práva vlastníka programu. Program **setuid** beží s prístupovými právami svojho vlastníka, kým program **setgid** má prístupové práva svojej skupiny a oba bity môžu byť nastavené podľa mechanizmu oprávnení.

Aj keď má proces priradené ďalšie prístupové práva, tieto práva sú kontrolované programom, ktorý dané práva vlastní. Programy **setuid** a **setgid** teda umožňujú užívateľom programované ovládanie prístupu, pri ktorom sú prístupové práva poskytované nepriamo. Program sa chová ako dôveryhodný subsystém a chráni prístupové práva užívateľa.

Aj keď tieto programy možno veľmi efektívne používať, existuje bezpečnostné riziko v prípade, že nie sú pozorne navrhnuté. Obzvlášť dôležitá je skutočnosť, že program nesmie nikdy vrátiť riadenie užívateľovi, ak ešte stále má prístupové práva svojho vlastníka, pretože to by umožnilo užívateľovi neobmedzené používanie prístupových práv vlastníka.

Poznámka: Z bezpečnostných dôvodov nepodporuje operačný systém volania programov **setuid** alebo **setgid** v rámci skriptu prostredia Shell.

Použitie S bitov pre ACL

ACL, ako napríklad NFS4, sa priamo nezaoberajú S bitmi. NFS4 ACL nešpecifikuje, ako by mohli byť tieto bity umiestnené ako súčasť ACL. Operačný systém AIX zaujal k tomuto problému taký prístup, že bity S sa budú používať počas vykonávania kontrol a budú doplňať prístupové kontroly súvisiace s NFS4 ACL. Nastavovať alebo nulovať bity S na objektoch súborového systému s ACL napríklad NFS4 môžete pomocou príkazu **chmod** poskytnutým s operačným systémom AIX.

Administratívne prístupové práva

Operačný systém poskytuje privilegované prístupové práva pre systémovú správu.

Systémové privilégia sú založené na ID užívateľa a skupiny. Užívatelia s efektívnym ID užívateľa alebo skupiny hodnoty 0 sú rozoznaní ako privilegovaní.

Procesy s efektívnymi ID užívateľov rovnými 0 sú známe ako procesy užívateľa s oprávneniami typu root a môžu:

- Čítať alebo zapisovať do akéhokoľvek objektu
- Volať všetky systémové funkcie
- Spúšťaním programov **setuid-root** vykonávať určité riadiace operácie subsystému.

Systém môžete riadiť dvomi typmi privilégií: privilégiom príkazu **su** a privilégiom programu **setuid-root**. Príkaz **su** umožňuje všetkým programom, ktoré vyvoláte, fungovať ako procesy užívateľa s oprávneniami typu root. Príkaz **su** je flexibilný spôsob na spravovanie systému, ale nie je príliš bezpečný.

Premena programu na program **setuid-root** znamená, že program bude program vo vlastníctve užívateľa s oprávneniami typu root s nastaveným bitom setuid. Program **setuid-root** poskytuje administratívne funkcie, ktoré môžu bez ohrozenia zabezpečenia vykonávať bežní užívatelia. Privilégium je teda obsiahnuté v programe a nie je priamo poskytnuté užívateľovi. Môže byť náročné uzatvoriť všetky potrebné administratívne funkcie do programov **setuid-root**, ale správcom systému to poskytuje väčšiu bezpečnosť.

Autorizácia prístupu

Keď sa užívateľ prihlási na konto (pomocou príkazov **login** alebo **su**), užívateľským procesom sú priradené ID užívateľa a ID skupiny priradené k danému kontu. Tieto ID určujú prístupové práva procesu.

Proces s ID užívateľa hodnoty 0 je známy ako *proces užívateľa s oprávneniami typu root*. Tieto procesy majú obvykle povolené všetky prístupové práva. Ak však proces užívateľa s oprávneniami typu root požaduje oprávnenie na spustenie programu, je prístup pridelený len v prípade, že je oprávnenie na spustenie pridelené aspoň jednému užívateľovi.

Autorizácia prístupu pre zoznamy ACL AIXC.

Vlastník informačného prostriedku je zodpovedný za riadenie prístupových práv. Prostriedky sú chránené pomocou *bitov oprávnení*, ktoré sú zahrnuté v režime objektu. Bity oprávnení definujú prístupové práva pridelené vlastníkovi objektu, skupine objektu a predvolenej triede others. Operačný systém podporuje tri rôzne režimy prístupu (na čítanie, zápis a spúšťanie), ktoré možno udeľovať samostatne.

V prípade súborov, adresárov, pomenovaných dátovodov a zariadení (špeciálnych súborov) je prístup autorizovaný nasledovne:

- Pre každú položku riadenia prístupu (ACE) v ACL sa porovná zoznam identifikátorov s identifikátormi procesu. V prípade zhody sú procesu udelené oprávnenia a obmedzenia definované pre danú položku. Logické zjednotenia pre obe oprávnenia sa vypočítajú pre každú zhodnú položku v zozname prístupových práv. Ak sa požadujúci proces nezhoduje so žiadnou z položiek na zozname ACL, budú mu udelené povolenia a obmedzenia predvolenej položky.
- Ak je požadovaný režim prístupu povolený (zahrnutý v zjednotení povolení) a nie je obmedzený (zahrnutý v zjednotení obmedzení), prístup sa pridelí. V opačnom prípade je prístup zakázaný.

Zoznam identifikátorov v zozname prístupových práv sa zhoduje s procesom, ak sa všetky identifikátory v zozname zhodujú so zodpovedajúcim typom efektívneho identifikátora pre požadujúci proces. Identifikátor typu USER sa zhoduje, ak zodpovedá ID efektívneho užívateľa procesu a identifikátor typu GROUP sa zhoduje, ak zodpovedá ID efektívnej skupiny procesu alebo jednému z ID doplnkových skupín. Napríklad, položka ovládania prístupu s nasledovným zoznamom identifikátorov:

```
USER:fred, GROUP:filozofi, GROUP:softverovi_programatori
```

by sa zhodovala s procesom s efektívnym ID užívateľa *fred* a množinou skupín:

```
filozofi, filantropi, softverovi_programatori, technici
```

nezhodovala by sa však s procesom s efektívnym ID užívateľa *fred* a množinou skupín:

```
filozofi, ucitelia, hardverovi_vyvojari, studenti
```

Všimnite si, že položka ovládania prístupu s nasledovným zoznamom identifikátorov by sa zhodovala s oboma procesmi:

```
USER:fred, GROUP:filozofi
```

Inými slovami, zoznam identifikátorov v položke ovládania prístupu predstavuje množinu podmienok, ktoré musia byť splnené, aby bol zadaný prístup pridelený.

Kontrola prístupových práv pre tieto objekty sa vykonáva na úrovni systémových volaní pri prvom prístupe k objektu. Pretože k objektom SVIPC sa pristupuje nezávisle, kontrola sa vykonáva pre každý prístup. V prípade objektov s názvami súborového systému je potrebné, aby bolo možné rozlíšiť názov aktuálneho objektu. Názvy sa rozlišujú buď relatívne (voči pracovnému adresáru procesu), alebo absolútne (voči koreňovému adresáru procesu). Celé rozlišovanie názvu začína vyhľadaním jedného z týchto adresárov.

Mechanizmus riadenia prístupu na základe uváženia užívateľa umožňuje efektívne ovládanie prístupu k informačným prostriedkom a poskytuje osobitnú ochranu dôvernosti a integrity informácií. Mechanizmy vlastníkom riadeného prístupu sú však len tak efektívne, ako ich užívatelia navrhnú. Všetci užívatelia by mali chápať princípy prideľovania a zakazovania prístupových práv a ich nastavovania.

Autorizácia prístupu pre zoznamy ACL NFS4

Každý užívateľ s privilégiom pre WRITE_ACL môže ovládať prístupové práva. Vlastník informačného prostriedku má vždy privilégium na WRITE_ACL. Pre súbory a adresáre so zoznamami ACL NFS4 sa prístup autorizuje takto:

- Zoznam ACE sa spracováva v poradí a na spracovanie sú určené len tie ACE, ktoré majú "who" (t.j. identitu) zhodnú so žiadateľom. Povoľovacie údaje žiadateľa sa počas spracovania ACE so špeciálnym who EVERYONE@ nekontrolujú.
- Každá ACE sa spracováva až do povolenia všetkých bitov žiadateľovho prístupu. Po povolení bitu sa tento už viac nezohľadňuje v spracovaní neskorších ACE.
- Ak je ktorýkoľvek bit zodpovedajúci žiadateľovmu prístupu odmietnutý, prístup bude odmietnutý a zvyšné ACE nebudú spracované.
- Ak neboli povolené všetky bity žiadateľovho prístupu a žiadna ďalšia ACE nezostala na spracovanie, prístup bude odmietnutý.

Ak je požadovaný prístup odmietnutý položkami ACE a žiadajúcim užívateľom je superužívateľ alebo kmeňový užívateľ, prístup bude vo všeobecnosti povolený. Všimnite si, že vlastník objektu má vždy povolené READ_ACL, WRITE_ACL, READ_ATTRIBUTES a WRITE_ATTRIBUTES. Bližšie informácie o algoritme autorizácie prístupu nájdete v “Zoznam riadenia prístupov NFS4” na strane 119.

Odstraňovanie problémov pre Access Control List

Nasledujúce informácie môžete použiť na odstránenie problémov zoznamu ACL.

NFS4 Access Control List v aplikácii zlyhaných objektov

Na odstraňovanie problémov s nastavovaním NFS4 ACL na objekte, napríklad súbore alebo adresári, môžete použiť návratový kód alebo zariadenie na sledovanie. Obe metódy používajú na zistenie príčiny problému príkazy **aclput** a **acledit**.

Použitie návratového kódu pri odstraňovaní problémov

Ak chcete zobraziť návratový kód, použite príkaz echo $? po spustení príkazu **aclput**. Nasledujúce zoznamy zobrazujú návratové kódy a ich vysvetlenia:

22 (EINVAL, definovaný v /usr/include/sys/errno.h)

Nasledujú možné príčiny pre tento kód:

- Neplatný textový formát v niektorom z týchto 4 polí.
- Veľkosť výstupu NFS4 ACL presahuje 64 KB.
- ACL sa použije pre súbor, ktorý už má minimálne jednu ACE s maskou ACE nastavenou na w (WRITE_DATA), ale nie p (APPEND_DATA) alebo p (APPEND_DATA), ale nie w (WRITE_DATA).
- ACL sa použije pre adresár, ktorý už má minimálne jednu ACE s maskou ACE nastavenou na w (WRITE_DATA), ale nie p (APPEND_DATA) alebo p (APPEND_DATA), ale nie w (WRITE_DATA) a príznak ACE fi (FILE_INHERIT).
- Existuje aspoň jeden ACE, ktorý má OWNER@ nastavený ako špeciálne **who** (**Identity**) a jedna alebo viaceré ACE masky c (READ_ACL), C (WRITE_ACL), a (READ_ATTRIBUTE) a A (WRITE_ATTRIBUTE) odmieta ACE typu d.

124 (ENOTSUP, definovaný v /usr/include/sys/errno.h)

Nasledujú možné príčiny pre tento kód:

- Špeciálna hodnota who nie je žiadnou z týchto troch hodnôt (OWNER@, GROUP@ alebo EVERYONE@) v jednej z položiek ACE.
- Existuje minimálne jedna položka ACE s typom ACE u (AUDIT) alebo l (ALARM).

13 (EACCES, definovaný v /usr/include/sys/errno.h)

Nasledujú možné príčiny pre tento kód:

- Nemáte povolenie čítať vstupný súbor obsahujúci NFS4 ACE.
- Nemáte povolenie hľadať rodičovský adresár cieľového objektu, pretože nemáte naň povolenie x (EXECUTE).
- Možno nemáte povolenie zapisovať alebo meniť ACL. Ak je už objekt priradený k NFS4 ACL, skontrolujte, či máte privilégium pre masku ACE C (WRITE_ACL).

Použitie sledovacieho zariadenia na odstraňovanie problémov

Ak chcete zistiť príčinu problému, môžete tiež vygenerovať správu sledovania. Nasledujúci scenár zobrazuje, ako použiť sledovanie na zistenie príčiny problému pomocou NFS4 ACL. Ak máte súbor /j2v2/file1 s týmto NFS4 ACL:

```
s:(EVERYONE@):  a       acC
```

a tento ACL sa nachádza vo vstupnom súbore input_acl_file:

```
s:(EVERYONE@):  a       rwxacC
```

Ak chcete odstrániť problémy pomocou sledovacieho zariadenia, postupujte nasledovne:

1. Pomocou nasledovných príkazov spustite sledovanie **aclput** a **trcrpt**:

```
$ trace -j  478 -o trc.raw
$->!aclput -i input_acl_file -t NFS4 /j2v2/file1
$ ->quit
$ trcrpt trc.raw > trc.rpt
```

2. Zanalyzujte správu o sledovaní. Keď sa použije ACL v súbore alebo adresári, kontroluje prístup na písanie alebo zmenu ACL a potom použije ACL. Súbor obsahuje riadky, podobné týmto:

```
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
```

   Druhý riadok obsahujúci chk_access exit uvádza, že prístup je povolený (rc = 0) na písanie ACL. Štvrtý riadok, ktorý obsahuje validate_acl a piaty riadok, ktorý obsahuje set_acl exit indikujú, že ACL nebolo úspešne použité (rc=22 indikuje **EINVAL**). Štvrtý riadok obsahujúci validate_acl uvádza, že v prvom riadku ACE (ace_cnt=1) sa vyskytol problém. Ak si pozriete prvú položku ACE, s:(EVERYONE@): a rwxacC), nenachádza sa tam žiadne **p** ako prístupová maska. Keď sa použije ACL, spolu s **p** sa vyžaduje aj **w**.

Odstraňovanie problémov s odmietnutím prístupu

Operácia systému súborov (napríklad zápis alebo čítanie) môže na objekte priradenom k NFS4 ACL zlyhať. Zvyčajne sa pri tom zobrazí chybová správa, ktorá však nemusí obsahovať dostatok informácií na určenie prístupového problému. Na zistenie prístupového problému môžete použiť zariadenie na sledovanie. Ak máte napríklad súbor /j2v2/file2 s nasledujúcim NFS4 ACL:

```
s:(EVERYONE@):  a       rwpx
```

Nasledujúci príkaz hlási chybu "Permission denied":

```
ls -l /j2v2/file2
```

Ak chcete odstrániť tento problém, postupujte nasledovne:

1. Sledovanie, ls -l /j2v2/file2 a trcrpt spustite pomocou nasledujúcich príkazov:

```
$ trace -j  478 -o trc.raw
$->!ls -l /j2v2/file2
$ ->quit
$ trcrpt trc.raw > trc.rpt
```

2. Zanalyzujte správu o sledovaní. Súbor obsahuje riadky, podobné týmto:

```
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
```

   Tretí riadok indikuje, že prístup je odmietnutý pre access mask = 128 (**0x80**), ktorý je len READ_ATTRIBUTES (pozrite si súbor /usr/include/sys/acl.h).

Prehľad auditu

Podsystém auditu umožňuje správcovi systému zaznamenať bezpečnostné informácie, ktoré môže zanalyzovať a získať tak informácie o potenciálnych a skutočných útokoch na bezpečnostnú politiku systému.

Podsystém auditu

Podsystém auditu má funkcie zisťovania, zhromažďovania a spracovania.

- “Zisťovanie udalostí auditu” na strane 128
- “Zhromažďovanie informácií o udalosti” na strane 128
- “Spracovanie informácií o protokole auditu” na strane 128

Systémový administrátor môže konfigurovať každú z týchto funkcií.

Zisťovanie udalostí auditu

Zisťovanie udalostí je distribuované v rámci súčasti TCB (Trusted Computing Base), v jadre (kód stavu supervízor) a v dôveryhodných programoch (kód stavu užívateľ). Auditovateľná udalosť predstavuje akýkoľvek výskyt udalosti týkajúcej sa bezpečnosti. Pod pojmom udalosť týkajúca sa bezpečnosti sa rozumie akákoľvek zmena stavu zabezpečenia systému a akýkoľvek pokus o narušenie alebo skutočné narušenie systémovej bezpečnostnej politiky pre ovládanie prístupu alebo politiky pre zabezpečenie kont, prípadne narušenie oboch týchto politík. Programy a moduly jadra, ktoré zisťujú auditovateľné udalosti, sú zodpovedné za oznamovanie týchto udalostí systémovému protokolovaču auditovania, ktorý je spustený ako súčasť jadra a možno k nemu pristupovať prostredníctvom podprogramu (v prípade auditovania dôveryhodného programu) alebo v rámci volania procedúry jadra (v prípade auditovania v stave supervízor). Zaznamenané informácie zahŕňajú názov auditovateľnej udalosti, úspech alebo neúspech udalosti a všetky ďalšie informácie týkajúce sa udalosti súvisiace s auditom bezpečnosti.

Konfigurácia zisťovania udalostí spočíva v povolení alebo zakázaní zisťovania udalostí a v určení, ktoré udalosti a pre ktorých užívateľov sa majú sledovať. Na aktiváciu zisťovania udalostí slúži príkaz **audit**, prostredníctvom ktorého možno povoliť alebo zakázať subsystém auditovania. Súbor /etc/security/audit/config obsahuje udalosti a užívateľov, ktorí sú spracovaní podsystémom auditu.

Zhromažďovanie informácií o udalosti

Zhromažďovanie informácií zahŕňa protokolovanie vybratých auditovateľných udalostí. Túto činnosť vykonáva protokolovač auditovania v jadre obsahujúci rozhranie pre systémové volania a volania procedúr v rámci jadra, ktoré zaznamenávajú auditovateľné udalosti.

Protokolovač auditovania je zodpovedný za vytvorenie úplného záznamu auditu skladajúceho sa z hlavičky auditu, ktorá obsahuje informácie spoločné pre všetky udalosti (názov udalosti, zodpovedný užívateľ, čas a návratový stav udalosti) a zo samotného protokolu auditu, ktorý obsahuje informácie o udalosti. Protokolovač auditovania pridáva každý nasledujúci záznam do protokolu predchádzajúceho auditu jadra, pričom tieto informácie môžu byť zapísané v jednom alebo oboch z nasledovných režimov:

Režim BIN
: Protokol sa zapisuje do striedajúcich sa súborov, čo umožňuje bezpečné a dlhodobé uloženie.

Režim STREAM
: Protokol sa zapisuje do cyklickej vyrovnávacej pamäte, ktorá sa synchrónne číta prostredníctvom pseudozariadenia auditovania. Režim STREAM ponúka okamžitú odozvu.

Zhromažďovanie informácií možno nastaviť pre obe fázy zhromažďovania, pre fázu zaznamenávania udalostí aj pre fázu spracovávania protokolu. Zaznamenávanie udalostí možno vyberať na základe jednotlivých užívateľov. Každý užívateľ má definovanú množinu udalostí auditovania, ktoré sa po ich nastaní zaprotokolujú do protokolu auditu. Vo fáze spracovávania sú režimy jednotlivo konfigurovateľné, takže administrátor môže využívať najvhodnejšie spracovanie údajov pre konkrétne prostredie. Ak sa príliš zmenší priestor systému súborov dostupný na protokolovanie, audit režimu BIN možno tiež nakonfigurovať na generovanie výstrahy.

Spracovanie informácií o protokole auditu

Operačný systém poskytuje niekoľko možností pre spracovanie protokolu auditu jadra. Pred prípadnou archiváciou protokolu auditu možno protokol režimu BIN komprimovať, filtrovať alebo formátovať pre výstup, prípadne vykonať kombináciu týchto možností. Kompresia sa vykonáva pomocou kódovania Huffman. Filtrovanie sa vykonáva výberom záznamov auditu pomocou syntaxe podobnej jazyku SQL (prostredníctvom príkazu **auditselect**), ktorý umožňuje selektívne prezerania a uchovávanie protokolu auditu. Formátovanie záznamov protokolu auditu slúži na kontrolu protokolu auditu, na generovanie pravidelných zostáv s informáciami o zabezpečení a na vytlačenie papierovej verzie protokolu auditu.

Protokol auditu režimu STREAM možno monitorovať v reálnom čase, čo umožňuje okamžité sledovanie ohrozenia. Konfigurácia týchto možností sa vykonáva pomocou samostatných programov, ktoré možno spustiť ako procesy typu

démon za účelom filtrovania protokolov režimu BIN alebo STREAM. Je potrebné vziať do úvahy, že niektoré filtrovacie programy sú viac prispôsobené na filtrovanie prvého alebo druhého režimu.

Konfigurácia podsystému auditovania

Subsystém auditovania obsahuje globálnu premennú stavu, ktorá indikuje, či je subsystém auditovania zapnutý. Okrem toho má každý proces lokálnu premennú stavu, ktorá určuje, či má podsystém auditovania zaznamenávať informácie o tomto procese.

Tieto dve premenné určujú, či sú modulmi a programami súčasti Trusted Computing Base (TCB) zisťované udalosti. Vypnutie auditovania pre konkrétny proces umožňuje danému procesu vykonávať svoje vlastné auditovanie a neumožňuje pritom obísť politiku zabezpečenia systémových kont. Povolenie dôveryhodnému programu, aby sa sám auditoval má za následok efektívnejšie zhromažďovanie informácií.

Kolekcia informácií podsystému auditovania

Kolekcia informácií adresuje výber udalostí a režimy protokolovania auditu jadra. Vykonáva sa prostredníctvom rutiny jadra poskytujúcej rozhrania pre protokolovanie informácií, ktoré využívajú súčasti TCB zisťujúce auditovateľné udalosti a konfiguračné rozhrania, ktoré využíva subsystém auditovania na riadenie rutiny pre protokolovanie auditu.

Protokolovanie auditu

Auditovateľné udalosti sú protokolované nasledovnými rozhraniami: stav užívateľa a stav supervízora. Užívateľská časť TCB využíva podprogram **auditlog** alebo **auditwrite**. Časť supervízora TCB využíva množinu volaní procedúr jadra.

V rámci každého záznamu vloží protokolovač auditovania udalostí pred informácie o udalosti hlavičku auditu. Táto hlavička identifikuje užívateľa a proces, pre ktorý je táto udalosť auditovaná, ako aj čas výskytu udalosti. Kód, ktorý zisťuje udalosť poskytne typ udalosti a návratový kód a stav, prípadne ďalšie informácie o udalosti. Informácie o udalosti sa skladajú z názvov objektov (napríklad súboru, ku ktorým bol zamietnutý prístup alebo tty použité v prípade neúspešných pokusov o prihlásenie), parametrov podprogramov a ďalších modifikovaných informácií.

Udalosti sú definované symbolicky, nie číselne. Znižuje sa tak pravdepodobnosť konfliktu názvov, bez použitia schémy registrácie udalostí. Pretože podprogramy sú auditovateľné a rozšíriteľná definícia jadra neobsahuje žiadne čísla SVC (switched virtual circuit), je zložité zaznamenávať udalosti podľa čísiel. Číselné mapovanie by muselo byť skontrolované a zaprotokolované vždy, keď by sa rozšírilo alebo predefinovalo rozhranie jadra.

Formát záznamu auditu

Záznamy auditu sa skladajú zo spoločnej hlavičky nasledovanej protokolmi auditu špecifickými pre udalosť auditu daného záznamu. Štruktúra hlavičiek je definovaná v súbore /usr/include/sys/audit.h. Formát informácií v protokoloch auditu je špecifický pre každú základnú udalosť a je zobrazený v súbore /etc/security/audit/events.

Informácie nachádzajúce sa v hlavičke obvykle zhromažďuje rutina protokolovača, aby sa zaistila ich presnosť. Informácie nachádzajúce sa v protokoloch auditu poskytuje kód, ktorý zisťuje udalosť. Protokolovač auditovania nevie, aká je štruktúra alebo sémantika protokolov auditu. Keď napríklad príkaz **login** zistí neúspešné prihlásenie, zaznamená túto udalosť na termináli, na ktorom k nej došlo a pomocou podrutiny **auditlog** zapíše záznam do protokolu auditu. Protokolovač auditovania, ktorý je súčasťou jadra, zaznamená do hlavičky informácie o subjekte (ID užívateľa, ID procesu, čas) a tieto informácie pridá k ďalším informáciám. Volajúca časť poskytne do hlavičky len názov udalosti a polia výsledkov.

Konfigurácia protokolovača auditu

Protokolovač auditovania je zodpovedný za vytvorenie úplného záznamu auditu. Je potrebné vybrať požadované udalosti auditu, ktoré sa majú protokolovať.

Výber udalostí auditu

Udalosti auditu môžu byť nasledovných typov:

Audit na proces
Za účelom efektívneho výberu udalostí procesu môže správca systému definovať triedy auditu. Trieda auditu predstavuje podmnožinu základných udalostí auditu v systéme. Triedy auditu poskytujú výhodné logické zoskupovanie základných udalostí auditu.

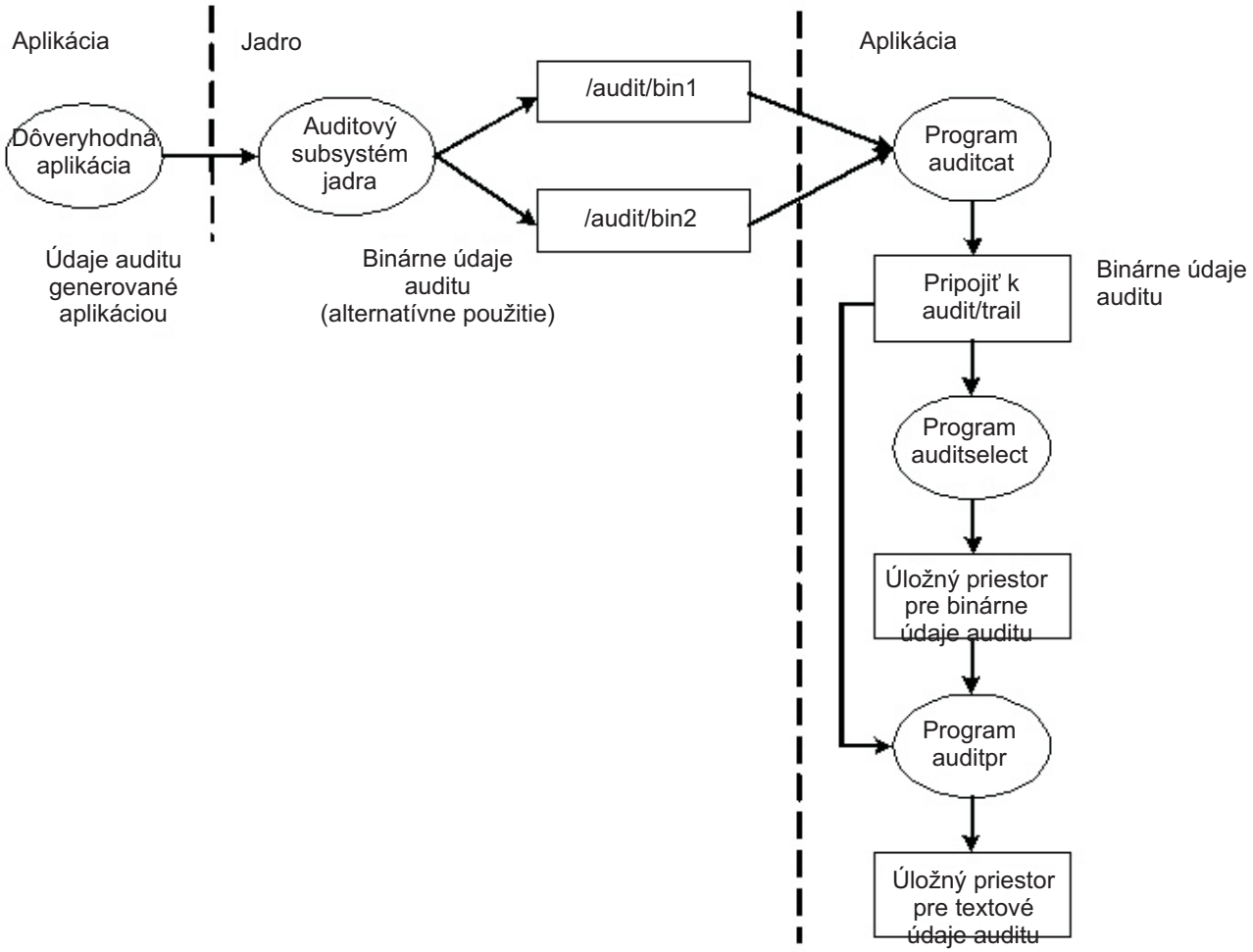
Pre každého užívateľa v systéme definuje správca systému množinu tried auditu určujúcu základné udalosti, ktoré možno zaznamenávať pre daného užívateľa. Každý užívateľom spustený proces je označený svojimi triedami auditu.

Audit na objekt
Operačný systém poskytuje auditovanie prístupov k objektom podľa názvov, to znamená, auditovanie konkrétnych objektov (obvykle súborov). Auditovanie objektov podľa názvu odstraňuje potrebu sledovania všetkých prístupov k objektom, ak je potrebné auditovať len niekoľko konkrétnych objektov. Okrem toho možno zadať režim auditu, takže zaznamenávané budú len prístupy zadaného režimu (na čítanie/zápis/spustenie).

Režimy protokolovania auditu jadra

Existujú dva režimy protokolovania jadra, BIN a STREAM, ktoré definujú, kde sa má zapisovať protokol auditu jadra. V prípade použitia režimu BIN musí byť protokolovaču auditovania jadra poskytnutý (pred začiatkom auditu) aspoň jeden deskriptor súboru, do ktorého sa majú pridávať záznamy.

Režim BIN pracuje na princípe zapisovania do dvoch striedajúcich sa súborov. Na začiatku auditovania sú jadru poskytnuté dva deskriptory súborov a odporúčaná maximálna veľkosť súborov bin. Jadro dočasne pozastaví volajúci proces a začne zapisovať záznamy auditu do prvého deskriptora súboru. Ak veľkosť prvého súboru bin dosiahne maximálnu veľkosť, a ak je druhý deskriptor súboru platný, jadro prepne na druhý súbor bin a opätovne aktivuje volajúci proces. Jadro pokračuje v zapisovaní do druhého súboru bin, až kým nie je znova zavolané a nie je mu predaný ďalší platný deskriptor súboru. Ak je v danom okamžiku druhý súbor bin plný, jadro prepne späť na prvý súbor bin a volajúci proces okamžite pokračuje. V opačnom prípade sa volajúci proces dočasne pozastaví a jadro pokračuje v zapisovaní záznamov do druhého súboru bin, až kým nie je plný. Spracovanie pokračuje týmto spôsobom až do vypnutia auditovania. Nasledovný obrázok ilustruje režim BIN auditu:

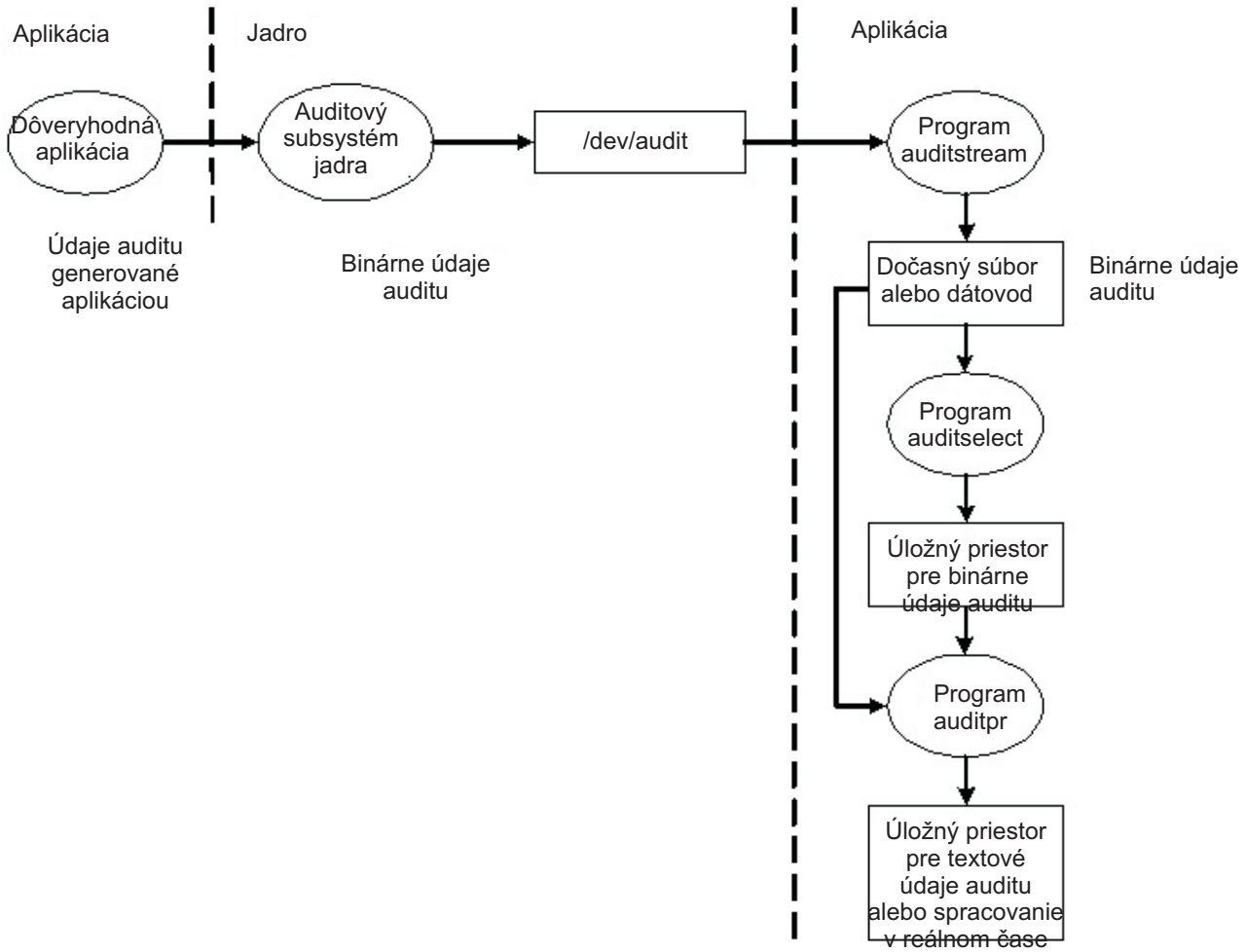


Obrázok 1. Proces režimu BIN auditu. Táto ilustrácia zobrazuje proces režimu BIN auditu.

Mechanizmus striedavého zapisovania do súborov bin slúži na zabezpečenie toho, aby subsystém auditovania mal počas spracovávania záznamov auditu vždy k dispozícii údaje na zápis. Keď subsystém auditovania prepína na druhý súbor bin, vyprázdni obsah prvého súboru bin do súboru trace. V čase prepnutia súborov bin je tak prvý súbor bin k dispozícii. Rozlišuje uloženie a analýzu údajov od generovania údajov. Program **auditcat** sa obvykle používa na čítanie údajov zo súboru bin, do ktorého práve jadro nezapisuje. Ak sa chcete presvedčiť, či systém nikdy nie je spúšťaný mimo priestoru pre protokolovanie auditu (výstup programu **auditcat**), môžete zadať parameter *freespace* v súbore /etc/security/audit/config. Ak má systém menej než tu zadaný rozsah 512-bajtových blokov, vygeneruje správu syslog.

Ak je povolený audit, parameter *binmode* v stanze start v /etc/security/audit/config by mal byť nastavený na panic. Parameter *freespace* v sekcii bin by mal byť nakonfigurovaný na hodnotu rovnajúcu sa najmenej 25% diskového priestoru vyhradeného na ukladanie protokolov auditu. Parametre *bytethreshold* a *binsize* by mali byť nastavené na veľkosť 65536 bajtov.

V režime STREAM zapisuje jadro záznamy do cyklickej vyrovnávacej pamäte. Ak jadro dosiahne koniec vyrovnávacej pamäte, jednoducho sa vráti späť na začiatok. Procesy čítajú informácie prostredníctvom pseudozariadenia s názvom /dev/audit. Keď nejaký proces otvorí toto zariadenie, pre tento proces sa otvorí kanál. Udalosti, ktoré sa majú čítať v rámci kanála je možné zadať ako zoznam tried auditu. Nasledovný obrázok ilustruje režim STREAM auditu:



Obrázok 2. Proces režimu STREAM auditu. Táto ilustrácia zobrazuje proces režimu STREAM auditu.

Hlavnou úlohou režimu STREAM je umožnenie priebežného čítania protokolu auditu, ktoré je potrebné pre monitorovanie ohrozenia v reálnom čase. Ďalšiu možnosť využitia predstavuje vytvorenie protokolu, ktorý je zapisovaný okamžite, bez akéhokoľvek spracovávania protokolu auditu, ako sa to deje v prípade zápisu na zapisovateľné médium.

Ďalšiu možnosť využitia režimu STREAM predstavuje zapisovanie toku údajov auditu do programu, ktorý ukladá informácie o audite na vzdialenom systéme umožňujúcom centrálne a rýchle spracovanie a zároveň chrániacom informácie o audite pred poškodením na pôvodnom hostiteľovi.

Spracovanie záznamov auditu

Príkazy **auditselect**, **auditpr** a **auditmerge** slúžia na spracovanie záznamov režimov BIN a STREAM auditu. Oba pomocné programy fungujú ako filtre, takže ich možno ľahko použiť na dátovodoch, čo je mimoriadne praktické pre audit režimu STREAM.

auditselect
Možno ho použiť na výber len určitých záznamov auditu s príkazmi podobnými SQL. Ak chcete vyberať napríklad len udalosti **exec()**, ktoré boli vygenerované užívateľom *afx*, napíšte:

```
auditselect -e "login==afx && event==PROC_Execute"
```

auditpr
Používa sa na konverziu binárnych záznamov auditu do formátu čitateľného pre človeka. Množstvo

zobrazených informácií záleží na príznakoch zadaných do príkazového riadka. Ak chcete získať všetky dostupné informácie, spustite príkaz **auditpr** nasledovne:

```
auditpr -v -hhelrtRpPTc
```

Ak zadáte príznak **-v**, okrem bežných informácií auditu poskytovaných jadrom pre každú udalosť sa zobrazí aj protokol auditu, ktorý je reťazcom konkrétnych udalostí (pozri súbor /etc/security/audit/events).

auditmerge

Tento príkaz sa používa na spájanie binárnych protokolov auditu. Jeho použitie je výhodné najmä vtedy, ak je potrebné skombinovať protokoly auditu z viacerých systémov. Príkaz **auditmerge** použije názvy protokolov v príkazovom riadku a odošle spojený binárny protokol na štandardný výstup, takže ešte musíte použiť príkaz**auditpr**, aby to bolo čitateľné. Napríklad príkazy **auditmerge** a **auditptr** možno spustiť nasledovne:

```
auditmerge trail.system1 trail.system2 | auditpr -v -hhelrRtpc
```

Použitie auditovacieho podsystému na rýchlu bezpečnostnú kontrolu:

Ak chcete monitorovať jeden podozrivý program bez nastavenia auditovacieho podsystému, môžete použiť príkaz **watch**. Tento príkaz zaznamenáva požadované alebo všetky udalosti generované zadaným programom.

Napríklad, ak chcete vidieť všetky udalosti **FILE_Open** pri spustení **vi /etc/hosts**, napíšte toto:

```
watch -eFILE_Open -o /tmp/vi.watch vi /etc/hosts
```

Súbor /tmp/vi.watch zobrazí všetky udalosti **FILE_Open** pre reláciu editora.

Výber udalostí

Výber udalostí musí udržiavať rovnováhu medzi nedostatočnými a prílišnými podrobnosťami.

Množina auditovateľných udalostí v systéme definuje, ktoré výskyty možno sledovať a podrobnosť vykonávaného auditovania. Auditovateľné udalosti musia zahŕňať udalosti týkajúce sa bezpečnosti, ako bolo skôr definované. Úroveň podrobností použitá v rámci definície auditovateľnej udalosti musí predstavovať kompromis medzi nedostatočnou úrovňou podrobností, ktorá správcovi sťažuje pochopenie vybratých informácií, a medzi príliš veľkým množstvom podorbností, ktoré má za následok prílišné zhromažďovanie informácií. Definícia udalostí využíva podobnosti medzi zistenými udalosťami. V nasledovnej diskusii bude *zistená udalosť* predstavovať ktorúkoľvek inštanciu auditovateľnej udalosti. Zadaná udalosť môže byť napríklad zistená v prípade rozličných miest. Principiálne to znamená, že zistené udalosti s podobnými vlastnosťami zabezpečenia sú vybraté ako jedna auditovateľná udalosť. V nasledovnom zozname je uvedená klasifikácia udalostí bezpečnostnej politiky:

- Udalosti subjektu
  - Vytvorenie procesu
  - Zrušenie procesu
  - Nastavenie atribútov zabezpečenia subjektu: ID užívateľa, ID skupiny
  - Skupina procesu, kontrolný terminál
- Udalosti objektu
  - Vytvorenie objektu
  - Vymazanie objektu
  - Otvorenie objektu (vrátane procesov ako objektov)
  - Zatvorenie objektu (vrátane procesov ako objektov)
  - Nastavenie atribútov zabezpečenia objektu: vlastník, skupina, zoznam prístupových práv
- Udalosti importu/exportu
  - Importovanie a exportovanie objektu
- Udalosti sledovania
  - Pridanie užívateľa, zmena atribútov užívateľa v databáze hesiel
  - Pridanie skupiny, zmena atribútov skupiny v databáze skupín

  - Prihlásenie užívateľa
  - Odhlásenie užívateľa
  - Zmena autentifikačných informácií užívateľa
  - Konfigurácia terminálu dôveryhodnej cesty
  - Konfigurácia autentifikácie
  - Administrácia auditovania: výber udalostí a protokolov auditu, zapnutie a vypnutie, definovanie tried auditovania užívateľov
- Všeobecné udalosti správy systému
  - Použitie privilégií
  - Konfigurácia súborového systému
  - Definícia a konfigurácia zariadení
  - Definícia parametrov konfigurácie systému
  - Normálne spustenie a vypnutie systému
  - Konfigurácia RAS
  - Iná konfigurácia systému
  - Spustenie podsystému auditu
  - Zastavenie podsystému auditu
  - Dotazovanie podsystému auditu
  - Resetovanie podsystému auditu
- Porušenie bezpečnosti (potenciálne)
  - Odmietnutia oprávnení prístupu
  - Zlyhania privilégií
  - Diagnosticky zistené chyby a systémové chyby
  - Pokus o zmenu TCB

Udalosti auditu:

Udalosť auditu je výskyt udalosti, ktorá je relevantná z hľadiska bezpečnosti, v systéme. Udalosť relevantná z hľadiska bezpečnosti môže byť zmena stavu zabezpečenia systému a pokus o narušenie politík riadenia prístupu alebo sledovania v systéme alebo ich úspešné narušenie. Programy a moduly jadra, ktoré zistia udalosti auditu, nahlásia tieto udalosti programu na zaznamenávanie systémového auditu spustenému v rámci jadra, ku ktorému môžete získať prístup prostredníctvom podrutiny (v prípade auditu dôveryhodného programu), alebo v rámci volania procedúry jadra (v prípade auditu stavu kontroly). Informácie, ktoré sa nahlásia v rámci udalosti auditu, zahrňujú názov udalosti, úspešnosť udalosti a ďalšie informácie o udalosti súvisiace s auditom zabezpečenia.

Ak chcete vykonať audit činnosti, musíte identifikovať príkaz alebo proces, ktorý iniciuje udalosť auditu a presvedčiť sa, či je udalosť uvedená v súbore /etc/security/audit/events pre váš systém. Priradenie udalostí auditu užívateľom možno zjednodušiť zlúčením podobných udalostí do tried auditu. Tieto triedy auditu sú definované v sekcii classes súboru /etc/security/audit/config.

V nasledujúcej tabuľke sú uvedené niektoré bežne používané udalosti auditu, ktoré sa môžu vyskytnúť v operačnom systéme AIX:

Tabuľka 11. Udalosti auditu

Užívateľ alebo systémové volanie	Udalosť auditu	Popis
fork	PROC_Create	Značí, že sa vytvára proces.
exit	PROC_Delete	Značí, že volajúci proces bol ukončený.
exec	PROC_Execute	Spustí nový program.
setuidx	PROC_RealUID	Nastaví identifikátor užívateľa pre proces.
	PROC_AuditID	
	PROC_SetUserIDs	
setgidx	PROC_RealGID	Nastaví ID skupiny pre proces.
accessx	FILE_Accessx	Značí prístupnosť súboru.
statacl	FILE_StatAcl	Získa informácie riadenia prístupu pre súbor.
revoke	FILE_Revoke	Zruší prístup k súboru pre všetky procesy.
frevoke	FILE_Frevoke	Zruší prístup k súboru pre iné procesy.
usrinfo	PROC_Environ	Zmení časť údajov užívateľských informácií.
sigaction	PROC_SetSignal	Značí akciu, ktorá sa má vykonať pri doručení určitého signálu do procesu, ktorý spustil podrutinu.
setrlimit	PROC_Limits	Riadi maximálnu spotrebu systémových prostriedkov.
nice	PROC_SetPri	Značí použitie funkcie nice.
setpri	PROC_Setpri	Nastaví pevnú prioritu pre procesy.
setpriv	PROC_Privilege	Zmení niektoré vektory privilégií pre procesy.
settimer	PROC_Settimer	Nastaví aktuálnu hodnotu pre určený systémový časovač.
adjtime	PROC_Adjtime	Zmení systémové hodiny.
ptrace	PROC_Debug	Sleduje spúšťanie iného procesu.
kill	PROC_Kill	Pošle signál do procesu alebo skupiny procesov.
setpgid	PROC_setpgid	Nastaví ID skupiny pre proces.
ld_loadmodule	PROC_Load	Načíta nový modul objektu do adresného priestoru procesu.
	PROC_LoadError	Značí, že sa nepodarilo načítať objekt.
setgroups	PROC_SetGroups	Zmení nastavenú súbežnú skupinu procesu.
sysconfig	PROC_Sysconfig	Zaznamená akciu v konfigurácii jadra alebo systému.
audit	AUD_It	Spustí alebo zastaví operáciu auditu. Taktiež zistí stav auditu.
auditbin	AUD_Bin_Def	Zmení systémové volanie auditbin.
auditevents	AUD_Events	Upraví udalosti.
auditobj	AUD_Objects	Upraví systémové volanie auditobj.
auditproc	AUD_Proc	Zistí alebo nastaví stav auditu pre proces.
acct	ACCT_Disable	Zakáže účtovanie v systéme.
	ACCT_Enable	Povolí účtovanie v systéme.
open a create	FILE_Open	Zavolá podrutinu **open**.
read	FILE_Read	Prečíta údaje z deskriptora súboru.
write	FILE_Write	Zapíše údaje do deskriptora súboru.
close	FILE_Close	Zatvorí otvorený deskriptor súboru.
link	FILE_Link	Vytvorí novú položku adresára pre objekt súborového systému.
unlink	FILE_Unlink	Odstráni objekt súborového systému.

Tabuľka 11. Udalosti auditu (pokračovanie)

Užívateľ alebo systémové volanie	Udalosť auditu	Popis
rename	FILE_Rename	Zmení názov objektu súborového systému.
chown	FILE_Owner	Zmení vlastníctvo súboru.
chmod	FILE_Mode	Zmení režim súboru.
fchmod	FILE_Fchmod	Zmení oprávnenia pre súbor v deskriptore súboru.
fchown	FILE_Fchown	Zmení vlastníctvo deskriptora súboru.
truncate	FILE_Truncate	Zmení dĺžku regulárnych súborov alebo objektu zdieľanej pamäte.
symlink	FILE_Symlink	Vytvorí symbolický odkaz.
pipe	FILE_Pipe	Vytvorí nepomenovaný dátovod.
mknod	FILE_Mknod	Vytvorí špeciálny súbor zariadenia alebo špeciálny súbor FIFO.
fcntl	FILE_Dupfd	Vytvorí duplikát deskriptoru súboru.
fscntl	FS_Extend	Rozšíri súborový systém.
mount	FS_Mount	Pripojí súborový systém k určenému adresáru.
umount	FS_Umount	Odpojí pripojený súborový systém.
chacl	FILE_Acl	Zmení zoznam riadenia prístupu (ACL) pre súbor.
fchacl	FILE_Facl	Zmení zoznam riadenia prístupu pre deskriptor súboru.
chpriv	FILE_Privilege	Nastaví zoznam riadenia privilégií pre cestu k súboru.
	FILE_Chpriv	Zmení zoznam riadenia privilégií.
	FILE_Fchpriv	Zmení zoznam riadenia privilégií pre deskriptor súboru.
chdir	FS_Chdir	Zmení aktuálny pracovný adresár.
fchdir	FS_Fchdir	Zmení aktuálny pracovný adresár prostredníctvom deskriptora súboru.
chroot	FS_Chroot	Zmení význam koreňového adresára (/) pre aktuálny proces.
rmdir	FS_Rmdir	Odstráni objekt adresára.
mkdir	FS_Mkdir	Vytvorí adresár.
utimes	FILE_Utimes	Zavolá podrutinu **utimes**.
stat	FILE_Stat	Zavolá podrutinu **stat**.
msgget	MSG_Create	Vytvorí front správ.
msgrcv	MSG_Read	Prijme správu z frontu správ.
msgsnd	MSG_Write	Pošle správu do frontu správ.
msgctl	MSG_Delete	Odstráni front správ.
	MSG_Owner	Zmení vlastníctvo a prístupové práva pre front správ.
	MSG_Mode	Zistí prístupové práva pre front správ.
semget	SEM_Create	Vytvorí sadu semaforov.
semop	SEM_Op	Zvýši alebo zníži hodnotu semaforov.
semctl	SEM_Delete	Odstráni sadu semaforov.
	SEM_Owner	Zmení vlastníctvo a prístupové práva sady semaforov.
	SEM_Mode	Zistí prístupové práva sady semaforov.
shmget	SHM_Create	Vytvorí nový segment zdieľanej pamäte.

Tabuľka 11. Udalosti auditu (pokračovanie)

Užívateľ alebo systémové volanie	Udalosť auditu	Popis
shmat	SHM_Open	Zavolá podrutinu **shmat** s voľbou **Open**.
shmat	SHM_Detach	Zavolá podrutinu **shmat** s voľbou **Detach**.
shmctl	SHM_Close	Zatvorí segment zdieľanej pamäte.
	SHM_Owner	Zmení vlastníctvo a prístupové práva pre segment zdieľanej pamäte.
	SHM_Mode	Zistí prístupové práva pre segment zdieľanej pamäte.
tcpip user level	TCPIP_config	Zaznamená zmeny v rozhraní TCP/IP.
	TCPIP_host_id	Zaznamená pokusy o zmenu názvu hostiteľa systému.
	TCPIP_route	Zaznamená zmeny v smerovacej tabuľke.
	TCPIP_connect	Zavolá podrutinu **connect**.
	TCPIP_data_out	Odoslané údaje.
	TCPIP_data_in	Prijaté údaje.
	TCPIP_set_time	Zaznamená pokus o zmenu systémového času cez sieť.
tcpip kernel level	TCP_ksocket	Zavolá služby jadra TCP/IP.
	TCP_ksocketpair	
	TCP_kclose	
	TCP_ksetopt	
	TCP_kbind	
	TCP_klisten	
	TCP_kconnect	
	TCP_kaccept	
	TCP_kshutdown	
	TCP_ksend	
	TCP_kreceive	
tsm	USER_Login	Prihlási užívateľa do systému.
	PORT_Locked	Značí, že port bol uzamknutý pre neplatné pokusy o prihlásenie.
	TERM_Logout	Odhlási užívateľa zo systému.
rlogind alebo telnetd	USER_Exit	Značí, že užívateľ je odhlásený.
usrck	USER_Check	Overí presnosť definície užívateľa.
	USRCK_Error	
logout	USER_Logout	Zastaví všetky procesy na porte.
chsec	PORT_Change	Značí zmenu hodnôt atribútov portu.
chuser	USER_Change	Zmení užívateľské atribúty.
rmuser	USER_Remove	Odstráni užívateľa.
mkuser	USER_Create	Vytvorí užívateľa.
setgroups	USER_SetGroups	Nastaví ID dodatočnej skupiny pre aktuálny proces.
setsenv	USER_SetEnv	Nastaví premennú prostredia.
su	USER_SU	Zmení identifikátor užívateľa priradený k relácii.
grpck	GROUP_User	Odstráni neexistujúcich užívateľov zo skupiny.
	GROUP_Adms	Odstráni neexistujúcich administračných užívateľov zo skupiny.

Tabuľka 11. Udalosti auditu (pokračovanie)

Užívateľ alebo systémové volanie	Udalosť auditu	Popis
chgroup	GROUP_Change	Zmení atribúty skupiny.
mkgroup	GROUP_Create	Vytvorí skupinu.
rmgroup	GROUP_Remove	Odstráni skupinu.
passwd	PASSWORD_Change	Zmení heslo užívateľa.
pwdadm	PASSWORD_Flags	Zmení heslo administrátora.
pwdck	PASSWORD_Check	Overí presnosť lokálnych autentifikačných informácií.
	PASWORD_Ckerr	
startsrc	SRC_Start	Spustí radič systémových prostriedkov.
stopsrc	SRC_Stop	Zastaví radič systémových prostriedkov.
addssys	SRC_Addssys	Pridá definíciu SRCsubsys do triedy objektov podsystému.
chssys	SRC_Chssys	Zmení definíciu podsystému v triede objektov podsystému.
addserver	SRC_Addserver	Pridá definíciu podriadeného servera do triedy objektov podriadeného servera.
chserver	SRC_Chserver	Zmení definíciu podriadeného servera v triede objektov podriadeného servera.
rmsys	SRC_Delssys	Odstráni definíciu podsystému z triedy objektov podsystému.
rmserver	SRC_Delserver	Odstráni definíciu podriadeného servera z triedy objektov typu Subserver.
enq	ENQUE_admin	Zaradí súbor do frontu.
qdaemon	ENQUE_exec	Naplánuje spustenie úloh vo fronte.
sendmail	SENDMAIL_Config	Smeruje poštu pre lokálne doručenie a doručenie v sieti.
	SENDMAIL_ToFile	
at	AT_JobAdd	Odstráni alebo pridá príkazy, ktorých spustenie bolo naplánované pomocou príkazu **at**.
	At_JobRemove	
cron	CRON_JobRemove	Odstráni alebo pridá príkazy, ktorých spustenie bolo naplánované pomocou príkazu **cron**.
	CRON_JobAdd	
	CRON_Start	Označuje začiatok úlohy **cron**.
	CRON_Finish	Označuje koniec úlohy **cron**.
nvload	NVRAM_Config	Značí prístup k pamäti NVRAM (Non-Volatile Random-Access Memory).
cfgmgr	DEV_Configure	Nakonfiguruje zariadenia.
chdev and mkdev	DEV_Change	Značí zmenu v zariadení.
mkdev	DEV_Create	Značí, že zariadenie bolo vytvorené.
	DEV_Start	Značí, že zariadenie bolo spustené.
installp	INSTALLP_Inst	Nainštaluje softvérové produkty, ktoré sú dostupné v kompatibilnom inštalačnom balíku.
	INSTALLP_Exec	
rmdev	DEV_Stop	Značí, že zariadenie bolo zastavené.
	DEV_Unconfigure	Značí, že bola zrušená konfigurácia zariadenia.
	DEV_Remove	Značí, že zariadenie bolo odstránené.
lchangelv, lextendlv a lreducelv	LVM_ChangeLV	Značí, že bol zmenený logický zväzok.
lchangepv, ldeletepv a linstallpv	LVM_ChangeVG	Značí, že bola zmenená skupina zväzkov.
lcreatelv	LVM_CreateLV	Značí, že bol do systému pridaný logický zväzok.
lcreatevg	LVM_CreateVG	Značí, že bola v systéme vytvorená skupina zväzkov.

Tabuľka 11. Udalosti auditu (pokračovanie)

Užívateľ alebo systémové volanie	Udalosť auditu	Popis
ldeletepv	LVM_DeleteVG	Značí, že bola skupina zväzkov odstránená zo systému.
rmlv	LVM_DeleteLV	Značí, že bol logický zväzok odstránený zo systému.
lvaryoffvg	LVM_VaryoffVG	Deaktivuje skupinu zväzkov.
lvaryonvg	LVM_VaryonVG	Aktivuje skupinu zväzkov.
Operácie s logickými zväzkami	LVM_AddLV	Pridá logický zväzok do existujúcej skupiny zväzkov.
	LVM_KDeleteLV	Odstráni logický zväzok z existujúcej skupiny zväzkov.
	LVM_ExtendLV	Rozšíri logický zväzok pridaním už nepriradených fyzických oddielov zo skupiny zväzkov.
	LVM_ReduceLV	Zmenší logický zväzok.
	LVM_KChangeLV	Zmení existujúci logický zväzok.
	LVM_AvoidLV	Zakazuje vykonávanie špecifických operácií na logickom zväzku.
Operácie s fyzickými zväzkami	LVM_MissingPV	Pridá chýbajúci fyzický zväzok do existujúcej skupiny zväzkov.
	LVM_AddPV	Pridá fyzický zväzok do existujúcej skupiny zväzkov.
	LVM_AddMissPV	Pridá chýbajúci fyzický zväzok do existujúcej skupiny zväzkov.
	LVM_DeletePV	Odstráni fyzický zväzok z existujúcej skupiny zväzkov.
	LVM_RemovePV	Zruší fyzický zväzok z existujúcej skupiny zväzkov.
	LVM_AddVGSA	Pridá oblasť stavu skupiny zväzkov (VGSA) do existujúceho fyzického zväzku.
	LVM_DeleteVGSA	Odstráni oblasť VGSA z existujúceho fyzického zväzku.
Operácie so skupinami zväzkov	LVM_SetupVG	Nastaví skupinu zväzkov definovaním logických zväzkov a určením informácií o oblasti VGSA a pamäti cache konzistentnosti zrkadlového zápisu (MWCC).
	LVM_DefineVG	Definuje skupinu zväzkov v jadre.
	LVM_KDeleteVG	Odstráni skupinu zväzkov z jadra.
Operácie zálohovania a obnovy	BACKUP_Export	Zaznamená priebeh operácie zálohovania.
	RESTORE_Import	Zaznamená priebeh operácie obnovy.
shell	USER_Shell	Zaznamená informácie o tty užívateľa.
reboot	USER_Reboot	Zaznamená udalosť reštartovania systému.
	PROC_Reboot	Zaznamená udalosť reštartovania procesu. Podrutina **reboot** reštartuje systém alebo zopakuje operáciu počiatočného zavedenia programu (IPL) v systéme.

Nastavenie auditovania

Táto procedúra zobrazuje ako nastaviť podsystém auditovania. Podrobnejšie informácie nájdete v konfiguračných súboroch uvedených v tomto postupe.

1. Zo zoznamu v súbore /etc/security/audit/events vyberte systémové činnosti (udalosti), ktoré sa majú auditovať. Ak ste pre aplikácie alebo pre prípony jadra pridali nové udalosti auditu, musíte upraviť súbor, aby sa nové udalosti pridali.

- Udalosť do tohto súboru pridáte, ak ste do aplikačného programu (pomocou subrutiny **auditwrite** alebo **auditlog**) alebo do prípony jadra (pomocou služieb jadra **audit_svcstart**, **audit_svcbcopy** a **audit_svcfinis**)vložili kód pre protokolovanie tejto udalosti.
- Uistite sa, že formátovacie inštrukcie pre každú novú udalosť auditu sú zahrnuté v súbore /etc/security/audit/events. Tieto špecifikácie umožňujú príkazu **auditpr** zapisovať protokol auditu pri formátovaní záznamov auditu.

2. Zoskupte vybraté udalosti auditu do množín podobných položiek nazývaných *triedy auditu*. Definujte tieto triedy auditu v sekcii tried súboru /etc/security/audit/config.
3. Triedy auditu priraďte jednotlivým užívateľom a udalosti auditu priraďte súborom (objektom), ktoré chcete auditovať, a to nasledovne:
   - Ak chcete priradiť triedy auditu konkrétnemu užívateľovi, pridajte riadok do sekcie užívateľov v súbore /etc/security/audit/config. Na pridanie tried auditu pre užívateľa môžete použiť príkaz **chuser**.
   - Ak chcete priradiť udalosti auditu objektu (údaje alebo spustiteľný súbor), pridajte sekciu pre daný súbor do súboru /etc/security/audit/objects.
   - Štandardné triedy auditu môžete pre nových užívateľov špecifikovať upravením súboru /usr/lib/security/mkuser.default. Tento súbor obsahuje atribúty užívateľov, ktoré sa použijú pri generovaní ID nových užívateľov. Napríklad použite triedu auditu general pre všetky ID nových užívateľov, nasledovným spôsobom:

     ```
     user:
         auditclasses = general
         pgrp = zamestnanci
         groups = zamestnanci
         shell = /usr/bin/ksh
         home = /home/$USER
     ```

     Ak chcete zaznamenať všetky udalosti auditu, zadajte triedu ALL. Keď to urobíte, dokonca aj na priemerne vyťaženom systéme sa budú generovať obrovské množstvá údajov. Obvykle je preto praktickejšie ohraničiť množstvo zaznamenávaných udalostí.
4. V súbore /etc/security/audit/config nastavte požadovaný typ zhromažďovania údajov, zhromažďovanie BIN, zhromažďovanie STREAM alebo obidve tieto metódy. Pre údaje auditu použite samostatný súborový systém, aby údaje auditu nesúperili o miesto na disku s inými údajmi. Zabezpečíte tak dostatok voľného miesta pre údaje auditu. Typ zhromažďovania údajov nakonfigurujte nasledovne:
   - Ak chcete nakonfigurovať zhromažďovanie typu BIN:
     a. Povoľte zhromažďovanie v režime BIN tak, že v začiatočnej stati nastavíte binmode = on.
     b. Upravte stať binmode pre konfiguráciu súborov pamäte a protokolu a zadajte cestu k súboru, ktorý obsahuje príkazy záložného spracovania režimu BIN. Predvolený súbor pre príkazy fázy spracovania údajov je súbor /etc/security/audit/bincmds.
     c. Uistite sa, že súbory bin auditu sú dostatočne veľké a nastavte parameter *freespace* tak, aby sa v prípade zaplnenia súborového systému zobrazila výstraha.
     d. Príkazy prostredia shell, ktoré spracovávajú súbory pamäte auditu v dátovode auditu zahrňte do súboru /etc/security/audit/bincmds.
   - Ak chcete nakonfigurovať zhromažďovanie typu STREAM:
     a. Povoľte zhromažďovanie v režime STREAM tak, že v začiatočnej stati nastavíte streammode = on.
     b. Do sekcie streammode zadajte cestu k súboru, ktorý obsahuje príkazy pre spracovanie údajov režimu STREAM. Predvolený súbor obsahujúci tieto informácie predstavuje súbor /etc/security/audit/streamcmds.
     c. Do súboru /etc/security/audit/streamcmds zahrňte príkazy používateľského prostredia, ktoré slúžia na spracovanie záznamov STREAM s pomocou dátovodu auditu.
5. Po dokončení všetkých potrebných zmien v konfiguračných súboroch môžete začať používať príkaz **audit start** na zapnutie podsystému auditu. Vygeneruje sa udalosť **AUD_It** s hodnotou 1.
6. Príkaz **audit query** použite na to, aby ste zistili, ktoré udalosti a objekty sú auditované. Vygeneruje sa udalosť **AUD_It** s hodnotou 2.
7. Príkaz **audit shutdown** použite na opätovnú deaktiváciu podsystému auditu. Vygeneruje sa udalosť **AUD_It** s hodnotou 4.

Generovanie všeobecného protokolu auditu:

Nasledujú príklady generovania generického protokolu auditu.

V tomto príklade predpokladajme, že správca systému chce použiť subsystém auditovania na monitorovanie veľkého viacužívateľského serverového systému. Nie je vykonaná žiadna priama integrácia do systému IDS, vo všetkých záznamoch auditu budú podozrivé udalosti vyhľadávané manuálne. Zaznamenáva sa len niekoľko dôležitých udalostí, aby množstvo generovaných údajov neprekročilo rozumnú veľkosť.

V rámci udalostí auditu sa budú zisťovať nasledovné udalosti:

FILE_Write
: Chceme vedieť o zápisoch súboru do konfiguračných súborov, takže táto udalosť bude použitá pri všetkých súboroch v strome /etc.

PROC_SetUserIDs
: Všetky zmeny užívateľských ID

AUD_Bin_Def
: Konfigurácia súborov bin auditu

USER_SU
: Príkaz **su**

PASSWORD_Change
: Príkaz **passwd**

AUD_Lost_Rec
: Upozornenie v prípade stratených záznamov

CRON_JobAdd
: nové úlohy cron

AT_JobAdd
: nové v úlohách

USER_Login
: Všetky prihlásenia

PORT_Locked
: Všetky uzamknutia terminálov v dôsledku príliš veľkého množstva neplatných pokusov

Nasleduje príklad generovania bežného protokolu auditu:

1. Nasledovným spôsobom nastavte zoznam kritických súborov, ktorých zmeny sa majú sledovať, napríklad všetky súbory v adresári /etc a v súbore objects pre ne nakonfigurujte sledovanie udalostí **FILE_Write**:
   ```
   find /etc -type f | awk '{printf("%s:\n\tw = FILE_Write\n\n",$1)}' >> /etc/security/audit/objects
   ```
2. Pomocou príkazu **auditcat** nastavte auditovanie v režime BIN. Súbor /etc/security/audit/bincmds obsahuje zápis podobný nasledovnému:
   ```
   /usr/sbin/auditcat -p -o $trail $bin
   ```
3. Upravte súbor /etc/security/audit/config a pridajte triedu pre udalosti, ktoré sa majú sledovať. Zadajte zoznam všetkých existujúcich užívateľov a špecifikujte pre nich triedu custom.
   ```
   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. Pridajte triedu auditu custom do súboru /usr/lib/security/mkuser.default, aby nové ID automaticky mali priradené správne volanie auditu:

```
user:
    auditclasses = custom
    pgrp = zamestnanci
    groups = zamestnanci
    shell = /usr/bin/ksh
    home = /home/$USER
```

5. Pomocou SMIT alebo príkazu **crfs** vytvorte nový systém súborov s názvom /audit. Systém súborov by mal byť dosť veľký, aby mohol obsahovať dve väzby a veľký protokol auditu.
6. Spustite voľbu príkazu **audit start** s pozrite si súbor /audit. Spočiatku by sa mali zobraziť dva súbory bin a prázdny súbor trail. Potom, ako ste systém chvíľu používali, by ste mali mať záznamy auditu v súbore trail, ktorý možno čítať s:

```
auditpr  -hhelpPRtTc -v | more
```

V tomto príklade sa používa len niekoľko udalostí. Ak chcete zaznamenávať všetky udalosti, môžete pre všetkých užívateľov zadať pre názov triedy hodnotu ALL. Táto akcia vygeneruje veľké množstvo údajov. Vhodné je pridať všetky udalosti týkajúce sa užívateľských zmien a zmien v privilégiách vo vašej triede custom.

Monitorovanie prístupu do súboru pre kritické súbory v reálnom čase:

Tieto kroky môžu byť použité pre monitorovanie prístupu do súboru pre kritické súbory v reálnom čase.

Vykonajte tieto kroky:

1. Nastavte zoznam kritických súborov, ktorých zmeny sa majú sledovať, napríklad všetky súbory v adresári /etc, a v súbore objects pre ne nakonfigurujte sledovanie udalostí **FILE_Write**:

```
find /etc -type f | awk '{printf("%s:\n\tw = FILE_Write\n\n",$1)}' >> /etc/security/audit/objects
```

2. Nastavte auditovanie v režime STREAM, aby sa zobrazil zoznam všetkých zápisov do súboru. (Tento príklad zobrazuje všetky zápisy do súboru na konzolu, avšak v prípade reálneho nasadenia môže byť vhodné použiť spracovanie, ktoré posiela udalosti do systému IDS na zisťovanie nežiaducich vniknutí do systému.) Súbor /etc/security/audit/streamcmds obsahuje zápis podobný nasledovnému:

```
/usr/sbin/auditstream | /usr/sbin/auditselect -e "event == FILE_Write" |
auditpr  -hhelpPRtTc -v > /dev/console &
```

3. V súbore /etc/security/audit/config nastavte auditovanie v režime STREAM a pridajte triedu pre udalosti zápisu do súboru. Následne nakonfigurujte všetkých užívateľov, ktorí majú byť auditovaní v rámci danej triedy:

```
start:
        binmode = off
        streammode = on

stream:
        cmds = /etc/security/audit/streamcmds

classes:
        filemon = FILE_write
```

```
users:
        root = filemon
        afx = filemon
        ...
```

4. Teraz spustite príkaz **audit start**. Všetky udalosti **FILE_Write** sa zobrazia na konzole.

Výber udalostí auditu:

Účelom auditu je zistenie činností, ktoré môžu oslabiť zabezpečenie vášho systému.

Ak nasledovné činnosti vykonáva neautorizovaný užívateľ, tieto činnosti ohrozujú bezpečnosť systému a predstavujú kandidátov na audit

- Vykonávanie činností v súčasti Trusted Computing Base
- Autentifikácia užívateľov
- Prístup k systému
- Zmena konfigurácie systému
- Úmyselné marenie činnosti systému auditovania
- Inicializácia systému
- Inštalácia programov
- Modifikácia kont
- Prenos informácií do systému a zo systému

Systém auditovania neobsahuje predvolenú množinu udalostí, ktoré sa majú auditovať. Sami si musíte vybrať udalosti alebo triedy udalostí, podľa vašich potrieb.

Ak chcete vykonať audit činnosti, musíte identifikovať príkaz alebo proces, ktorý iniciuje udalosť auditu a presvedčiť sa, či je udalosť uvedená v súbore /etc/security/audit/events pre váš systém. Následne musíte pridať udalosť do príslušnej triedy v súbore /etc/security/audit/config alebo do sekcie object v súbore /etc/security/audit/objects. V súbore /etc/security/audit/events vo vašom systéme nájdete zoznam udalostí auditu a pokyny pre formátovanie protokolu. Popis zápisu a používania formátov udalostí auditu nájdete v popise príkazu **auditpr**.

Po výbere udalostí, ktoré sa majú auditovať, je potrebné zlúčiť podobné udalosti do tried auditu. Triedy auditu sú následne priradené užívateľom.

Výber tried auditu

Priradenie udalostí auditu užívateľom možno zjednodušiť zlúčením podobných udalostí do tried auditu. Tieto triedy auditu sú definované v sekcii classes súboru /etc/security/audit/config.

Medzi typické triedy auditu patria nasledovné:

general
: Udalosti, ktoré menia stav systému a autentifikáciu užívateľa. Pokusy auditu o obchádzanie systémových prvkov pre ovládanie prístupu.

objects Prístup pre zápis do súborov konfigurácie zabezpečenia.

kernel Udalosti v triede kernel sú generované funkciami jadra pre riadenie procesov.

Nasleduje príklad sekcie v súbore /etc/security/audit/config:

```
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
```

Výber metódy pre zhromažďovanie údajov auditu

Výber metódy zhromažďovania údajov závisí na ďalšom spôsobe využívania údajov auditu. Ak potrebujete dlhodobo uložiť veľké množstvo údajov, vyberte zhromažďovanie BIN. Ak chcete údaje spracovávať v čase zhromažďovania, vyberte zhromažďovanie STREAM. Ak potrebujete dlhodobé uloženie a okamžité spracovanie, vyberte obe metódy. Nasledujú popisy každej z týchto metód:

Zhromažďovanie BIN
: Umožňuje dlhodobé uloženie veľkého protokolu auditu. Záznamy auditu sa zapisujú do súboru, ktorý slúži ako dočasné úložné miesto. Po zaplnení súboru začne subsystém auditovania zapisovať údaje do druhého súboru bin. Údaje z plného súboru medzitým spracuje démon **auditbin** a záznamy sa zapíšu do protokolu auditu.

Zhromažďovanie STREAM
: Umožňuje spracovávanie údajov auditu v čase zhromažďovania údajov. Záznamy auditu sú zapisované do cyklickej vyrovnávacej pamäte v jadre a čítané pomocou zariadenia /dev/audit. Záznamy auditu možno zobraziť, vytlačiť vo forme papierového protokolu auditu alebo prostredníctvom príkazu **auditcat** skonvertovať na záznamy bin.

Auditovanie oddielu pracovného zaťaženia

V prostredí WPAR sú k dispozícii tri typy auditovania: globálne, systémové a auditovanie z globálneho.

Môžete povoliť auditovanie v globálnom WPAR, vnútri WPAR alebo oboje. Konfigurácia auditu pre systémový WPAR a globálny WPAR je podobná konfigurácii v prostredí s výnimkou wpar. Môžete spustiť auditovanie globálneho WPAR pre oddiely WPAR systému a aplikácií.

Poznámka: Auditovanie pre oddiely WPAR aplikácií nemôžete spustiť zvnútra oddielu WPAR, ale môžete ho spustiť pomocou auditovania globálneho WPAR.

Auditovanie globálneho WPAR pomáha administrátorom globálneho systému auditovať oddiely WPAR z globálneho systému. Administrátor globálneho systému môže kontrolovať úroveň auditovania pre každý oddiel WPAR z jediného miesta zadaním tried, ktoré sa majú auditovať pre každý oddiel WPAR v globálnom súbore /etc/security/audit/config.

Pridaním odseku WPARS do súboru /etc/security/audit/config môže administrátor globálneho systému poskytnúť zoznam tried, ktoré sa majú auditovať pre WPAR. napríklad:

```
WPARS:
<wpar_name> = <auditclass>, ... <auditclass>
```

V predchádzajúcom príklade musí byť <wpar_name> názvom WPAR systému a každý parameter triedy auditu by mal byť definovaný v odseku tried.

Ak chcete konfigurovať auditovanie oddielu WPAR testwpar triedami general, tcpip a lvm, do súboru /etc/security/audit/config pridajte nasledujúci odsek:

```
WPARS:
testwpar = general,tcpip,lvm
```

Administrátor globálneho systému môže spustiť a zastaviť auditovanie na WPAR pomocou príkazu **audit** a zadaním nasledujúceho názvu WPAR:

```
audit start -@ <wparname1> -@ <wparname2> ...
audit shutdown -@ <wparname1> -@ <wparname2> ...
```

Objekty WPAR môžete auditovať z globálneho prostredia zadaním absolútnych ciest do objektov, ktoré chcete auditovať. Napríklad, ak chcete definovať udalosti auditu pre súbor /wpars/wpar1/etc/security/passwd, do súboru /etc/security/audit/objects v systéme AIX, ktorý je hostiteľom oddielu WPAR, pridajte nasledujúci odsek:

```
/wpars/wpar1/etc/security/passwd:
     r = "WPAR1_PASSWD_RD"
     w = "WPAR1_PASSWD_WR"
```

Tento predchádzajúci odsek je analyzovaný v čase spustenia auditu (-@ <wpar1>), aby sa zaplo auditovanie objektov pre objekt /etc/security/passwd oddielu wpar1. Tieto atribúty vygenerujú auditovaciu udalosť WPAR1_PASSWD_RD pri každom prečítaní súboru /wpars/wpar1/etc/security/passwd. Tieto atribúty vygenerujú auditovaciu udalosť WPAR1_PASSWD_WR aj pri každom otvorení súboru za účelom zapisovania.

Poznámka: Skôr ako povolíte auditovanie oddielu WPAR z globálneho prostredia, musíte povoliť auditovanie pre globálne prostredie.

Príkaz **auditpr** sa používa na vygenerovanie správy auditu, ktorá zobrazuje názov WPAR. napríklad:

```
auditpr -v < /audit/trail
```

Auditovanie v prostredí NFS

Podsystém auditovania systému AIX podporuje auditovanie pripojených súborových systémov. Konfigurácia pripojeného súborového systému na klientovi je podobná lokálnemu súborovému systému. Operácie auditovania na auditovateľných pripojených objektoch sú podobné operáciám na lokálnych objektoch, ako je popísané v časti Prehľad auditovania. Správanie auditovania na klientovi a serveri pre pripojené súborové systémy je popísané v tejto téme neskôr.

Auditovanie na klientovi NFS

Všetky operácie spustené na auditovateľných objektoch, ktoré sa nachádzajú na súborových systémoch pripojených klientom, sa protokolujú na klientovi. Toto platí v prípade, že server NFS alebo žiadni klienti NFS nevykonávajú žiadne operácie s týmito objektmi, alebo je na klienti povolené auditovanie s úplnou cestou.

Bližšie informácie nájdete na stránke manuálu pre príkaz **audit**. Ak nie je povolené auditovanie s úplnou cestou a súbor bol upravený serverom alebo klientmi, následné auditovanie bude nepredvídateľné. Toto správanie je možné opraviť reštartovaním auditovania na klientovi. Ak je súborový systém pripojený na viacerých klientoch, odporúča sa, že vykonáte auditovanie operácií na serveri, aby ste získali presný protokol udalostí, alebo povolíte auditovanie s úplnou cestou na klientovi.

Poznámka: Konfigurácia podsystému auditu nepodporuje použitie súborového systému protokolov z auditu ako pripojeného súborového systému NFS.

Auditovanie na serveri NFS

Všetky operácie, vykonávané na pripojenom súborovom systéme klientom aj serverom, sa protokolujú na serveri NFS.

Obmedzenia na strane servera

- Ak na server nie je odoslaná žiadna operácia, vykonaná klientom NFS (buď v dôsledku ukladania NFS do pamäte cache alebo v dôsledku inherentnej architektúry NFS), takáto operácia nebude serverom auditovaná.

  Napríklad: Po pripojení súborového systému je serverom auditovaná len prvá operácia čítania, vykonaná na súbore. Následné operácie čítania sa na serveri neprotokolujú. Toto sa vzťahuje na operácie čítania, vykonávané na súboroch, odkazoch a adresároch.
- Operácie vykonané klientom sa protokolujú na serveri ako **nfsd** a ako meno užívateľa je uvedené **root**.

Príklad

Súborový systém s názvom *File_System* je pripojený na klientovi príkazom **mount server:/File_system /mnt**. Ak musí byť súbor s názvom *A* v súborovom systéme *File_System* auditovaný na serveri, /File_system/A musí byť nakonfigurovaný v konfiguračných súboroch auditu.

Ak sa rozhodnete súbor *A* v súborovom systéme *File_System* auditovať na klientovi, /mnt/A musí byť nakonfigurovaný tak, aby bol auditovaný na klientovi.

Ak je súbor *A* nakonfigurovaný tak, aby bol auditovaný na serveri aj na klientovi, operácie, vykonané serverom aj klientom na súbore *A*, sa auditujú a protokolujú na serveri a operácie, vykonané klientom, sa protokolujú na klientovi.

Každá operácia, vykonaná klientom na súbore *A*, je na serveri zaprotokolovaná ako démon **nfsd** a nie ako názov operácie alebo príkazu.

Lightweight Directory Access Protocol

Lightweight Directory Access Protocol (LDAP) definuje štandardnú metódu prístupu a aktualizácie informácií v adresári (databáze) lokálne alebo vzdialene v modeli servera klienta.

Tento protokol je optimalizovaný na čítanie, prehliadanie a hľadanie adresárov a pôvodne bol vyvinutý ako lightweight front-end k X.500 Directory Access Protocol. Metódu LDAP používa klaster hostiteľov na zabezpečenie centralizovanej bezpečnostnej autentifikácie, ako aj prístupu k informáciám o užívateľoch a skupinách. Tieto funkcie sú určené na používanie v klastrovom prostredí pre uchovanie spoločných informácií autentifikácie, užívateľov a skupín v klastri.

Objekty sú v LDAP uložené v hierarchickej štruktúre známej ako adresárový informačný strom (DIT). Dobrý adresár sa začína štrukturálnym usporiadaním DIT, ktorý je potrebné starostlivo vytvoriť ešte pred implementáciou LDAP ako prostriedku autentifikácie.

Zavádzací modul autentifikácie LDAP

Bezpečnostný subsystém v rámci LDAP je implementovaný ako zavádzací modul autentifikácie LDAP. Koncepčne sa podobá ostatným zavádzacím modulom, napríklad NIS, DCE a KRB5. Zavádzacie moduly sú definované v súbore /usr/lib/security/methods.cfg.

Zavádzací modul LDAP poskytuje v celom protokole LDAP funkcie autentifikácie užívateľa a centrálnej správy užívateľov a skupín. Užívateľ definovaný na serveri LDAP môže byť nakonfigurovaný na prihlásenie do klienta LDAP, aj keď nie je lokálne zadefinovaný.

Zavádzací modul LDAP AIX je plne integrovaný do operačného systému AIX. Po autentifikácii LDAP je zavádzací modul zapnutý na poskytovanie užívateľských a skupinových informácií, vysoko úrovňových API, príkazov a práce nástrojov riadenia systému zvyčajným spôsobom. Pre väčšinu príkazov vyššej úrovne sa zaviedol príznak **-R**, ktorý sa používa v rôznych zavádzacích moduloch. Napríklad na vytvorenie užívateľa LDAP s menom *joe* na klientskom počítači použite nasledovný príkaz:

```
mkuser -R LDAP joe
```

Poznámka: I keď infraštruktúra LDAP môže podporovať neobmedzený počet užívateľov v skupine, v jednej skupine bolo vytvorených maximálne 25 000 užívateľov a boli testované rôzne operácie pre tú skupinu. Niektoré z historických rozhraní POSIX nemusia vrátiť úplné informácie pre skupinu. Ak sa chcete o týchto obmedzeniach dozvedieť viac, pozrite si dokumentáciu k príslušnému aplikačnému programovému rozhraniu (API).

Autentifikácia na báze LDAP:

Na rôznych entitách existujú limity ako súčasť LDAP autentifikácie na AIX.

Podotýkame, že infraštruktúra LDAP sama osebe nešpecifikuje pre obsah databáz nijaké obmedzenia. Táto časť však dokumentuje výsledky získané na základe skúšobných konfigurácií zvolených s ohľadom na obmedzenia. Boli testované nasledujúce obmedzenia vzhľadom na autentifikáciu LDAP v operačnom systéme AIX:

Celkový počet užívateľov: Bolo vytvorených maximálne 500 000 užívateľov v jednom systéme a simultánna autentifikácia bola testovaná pre stovky užívateľov.

Celkový počet skupín:V jedinom systéme bolo vytvorených a testovaných až 500 skupín.

Maximálny počet užívateľov v skupine: Bolo vytvorených maximálne 25 000 užívateľov a boli testované rôzne operácie pre tú skupinu.

Niektoré z historických rozhraní POSIX nemusia vrátiť úplné informácie pre skupinu. Ak sa chcete o týchto obmedzeniach dozvedieť viac, pozrite si dokumentáciu k príslušnému aplikačnému programovému rozhraniu (API). Okrem toho, vyššie uvedené hodnoty sú založené na uskutočnenom testovaní. Nevylučujú možnosť, že - za predpokladu, že sú k dispozícii potrebné prostriedky - niekto môže nakonfigurovať systémy s oveľa väčším počtom používateľov a skupín.

Konfigurácia servera bezpečnostných informácií IBM Tivoli Directory Server:

Ak chcete systém nakonfigurovať ako server bezpečnostných informácií LDAP, ktorý bude poskytovať autentifikačné informácie a informácie o užívateľoch a skupinách prostredníctvom protokolu LDAP, musíte najskôr nainštalovať server LDAP a klientske balíky.

Ak sa vyžaduje protokol SSL (Secure Sockets Layer), musíte nainštalovať aj balík **GSKitV7** - v prípade IBM Tivoli Directory Server, verzia 6.2, alebo staršej verzie, alebo balík GSKitV8 - v prípade IBM Tivoli Directory Server, verzia 6.3, alebo novšej verzie. Administrátor systému musí vytvoriť kľúč pomocou príkazu na správu kľúčov balíka GSKit. Týmto príkazom je príkaz **gsk7ikm** v balíku GSKitV7 alebo príkaz **ikeyman** v balíku GSKitV8. Bližšie informácie o konfigurácii servera pre používanie SSL nájdete v téme Zabezpečená komunikácia pomocou SSL.

Spustením príkazu **mksecldap** nakonfigurujte server. Príkaz **mksecldap** vytvorí server LDAP a jeho koncovú databázu s názvom *ldapdb2*, zavedie na server LDAP informácie o užívateľoch a skupinách z lokálneho hostiteľa a nastaví jedinečné meno (DN) a heslo pre administrátora servera LDAP. Voliteľne môže nastaviť protokol SSL pre komunikáciu medzi klientom a serverom. Príkaz **mksecldap** taktiež pridá položku do súboru /etc/inittab na spustenie servera LDAP po každom reštartovaní.

Užívatelia a skupiny systému AIX sa ukladajú na server LDAP použitím niektorej z nasledujúcich schém:

Schéma systému AIX
Obsahuje triedy objektov aixAccount a aixAccessGroup. Táto schéma poskytuje kompletnú množinu atribútov pre užívateľov a skupiny systému AIX.

Schéma RFC 2307
Obsahuje triedy objektov posixAccount, shadowAccount a posixGroup a používajú ju rôzne adresárové produkty. Schéma RFC 2307 definuje len malú podmnožinu atribútov, ktoré používa systém AIX.

Schéma RFC2307AIX
Zahŕňa triedy objektov posixAccount, shadowAccount a posixGroup plus triedy objektov aixAuxAccount a aixAuxGroup. Triedy objektov aixAuxAccount a aixAuxGroup poskytujú atribúty, ktoré používa systém AIX, ale nie sú definované schémou RFC 2307.

Dôrazne sa odporúča pre užívateľov a skupiny používať typ schémy RFC2307AIX. Typ schémy RFC2037AIX je úplne v súlade so štandardom RFC 2307 s ďalšími atribútmi podporujúcimi viac funkcií na správu užívateľov v systéme AIX. Server IBM Tivoli Directory Server s konfiguráciou schémy RFC2307AIX podporuje nielen klientov LDAP systému AIX, ale aj iných klientov LDAP systémov UNIX a Linux, ktorí sú v súlade so štandardom RFC 2307.

Všetky informácie o užívateľoch a skupinách sú uložené do spoločného stromu systému AIX (prípona). Štandardnou príponou je "cn=aixdata". Príkaz **mksecldap** akceptuje užívateľom zadanú príponu prostredníctvom príznaku **-d**. Názvy vytvorených podstromov pre užívateľov, skupiny, identifikátory, atď. sa riadi konfiguračným súborom sectoldif.cfg. Bližšie informácie nájdete v súbore sectoldif.cfg.

Strom systému AIX je chránený zoznamom riadenia prístupu (ACL). Štandardný zoznam ACL udeľuje privilégium správcu len entite uvedenej ako správca s voľbou príkazu **-a**. Identite proxy môže byť udelené ďalšie privilégium, ak budú použité voľby príkazu **-x** a **-X**. Použitie týchto volieb vytvára identitu proxy a konfiguruje prístupové privilégium podľa definície v súbore /etc/security/ldap/proxy.ldif.template. Vytvorenie identity proxy umožňuje klientom LDAP pripojiť sa k serveru bez použitia identity administrátora, čo obmedzuje oprávnenia administrátora klienta na server LDAP.

Príkaz **mksecldap** môžete spustiť aj na serveri LDAP, ktorý je nakonfigurovaný na iné účely, napríklad na vyhľadanie informácií o identifikátoroch užívateľov. V tomto príklade, príkaz **mksecldap** pridá strom systému AIX a naplní ho bezpečnostnými informáciami systému AIX pre existujúci server LDAP. Tento strom je chránený zoznamom riadenia prístupu (ACL) nezávisle od iných existujúcich stromov.

Poznámka: Pred spustením príkazu **mksecldap** a rozšírením servera na server bezpečnostných informácií systému AIX by ste mali zálohovať existujúci server LDAP.

Po úspešnom nakonfigurovaní servera bezpečnostných informácií LDAP môžete toho istého hostiteľa nakonfigurovať ako klienta na správu užívateľov a skupín LDAP a povolenie užívateľom LDAP prihlásiť sa na tento server.

Ak nebude nastavenie servera bezpečnostných informácií LDAP úspešné, nastavenie môžete zrušiť, keď príkaz **mksecldap** spustíte s príznakom **-U**. Tým sa súbor ibmslapd.conf (alebo slapd.conf alebo slapd32.conf) obnoví do stavu pred nastavením. Predtým ako sa znovu pokúsite spustiť príkaz **mksecldap**, najprv spustite po každom neúspešnom pokuse o nastavenie príkaz **mksecldap** s príznakom **-U**. V opačnom prípade môžu v konfiguračnom súbore zostať zbytočné informácie o inštalácii, ktoré spôsobia zlyhanie ďalšej inštalácie. Toto bezpečnostné opatrenie nemá vplyv na databázu a jej údaje, pretože databáza mohla existovať už pred spustením príkazu **mksecldap**. Ak bola databáza vytvorená pomocou príkazu **mksecldap**, odstráňte ju manuálne. Ak príkaz **mksecldap** pridal údaje do už existujúcej databázy, rozhodnite sa, aké kroky sa majú vykonať na obnovenie po neúspešnom pokuse o inštaláciu.

Súvisiace koncepty:

Komunikácia zabezpečená pomocou SSL
Podľa toho aký typ autentifikácie sa používa medzi serverom a klientom LDAP, heslá sa budú odosielať buď v šifrovanom formáte (unix_auth) alebo ako čitateľný text (ldap_auth). Na ochranu proti ohrozeniu bezpečnosti pri zasielaní dokonca aj šifrovaných hesiel cez sieť alebo v niektorých prípadoch cez internet použite SSL (Secure Socket Layer). AIX poskytuje balíky pre SSL, ktoré poskytujú bezpečnú komunikáciu medzi adresárovými servermi a klientmi.

Súvisiace informácie:

Príkaz **mksecldap**

Nastavenie klienta LDAP:

Ak si želáte nastaviť klienta tak, aby na autentifikáciu a správu informácií o používateľoch/skupinách používal protokol LDAP, presvedčte sa, či má každý klient nainštalovaný balík klienta LDAP. Ak sa vyžaduje protokol Secure Sockets Layer (SSL), musíte nainštalovať softvér GSKit, vytvoriť kľúč a musíte do tohto kľúča pridať certifikát kľúča SSL servera LDAP.

Podobne ako pri nastavení servera LDAP možno vykonať nastavenie klienta pomocou príkazu **mksecldap**. Aby mohol klient komunikovať so serverom LDAP security information server, počas inštalácie je potrebné zadať názov servera. DN väzby a heslo servera sa vyžadujú aj na klientsky prístup k stromu AIX na serveri. Príkaz **mksecldap** uloží DN väzby servera, heslo, názov servera, AIX DN stromu na serveri, cestu a heslo kľúča SSL a ďalšie atribúty konfigurácie do súboru /etc/security/ldap/ldap.cfg.

Príkaz **mksecldap** uloží heslo väzby a heslo kľúča SSL (ak konfigurujete protokol SSL) do súboru /etc/security/ldap/ldap.cfg v šifrovanom formáte. Šifrované heslá sú špecifické pre systém a môžu byť používané iba démonom **secldapclntd** v systéme, kde sú generované. Démon **secldapcIntd** môže použiť nešifrované alebo šifrované heslo zo súboru /etc/security/ldap/ldap.cfg.

Počas inštalácie klienta je možné v príkaze **mksecldap** zadať viacero serverov. V tom prípade sa klient snaží spojiť so servermi v zadanom poradí a pripojí sa na prvý server, na ktorý je to možné. Ak sa medzi klientom a serverom vyskytne chyba pripojenia, bude zadaná požiadavka o opätovné pripojenie pomocou tej istej logiky. Model zabezpečenia protokolu LDAP nepodporuje referencie. Replikované servery je potrebné synchronizovať.

Klient komunikuje so serverom bezpečnostných informácií LDAP prostredníctvom démona na strane klienta (**secldapcIntd**). Ak je na klientovi povolený zavádzací modul LDAP, vysoko úrovňové príkazy budú pre užívateľov

definovaných v LDAP smerované do démona cez API knižnice. Démon udržuje pamäť cache požadovaných položiek LDAP. Ak nie je požiadavka z pamäte cache uspokojená, démon dotazuje server, aktualizuje pamäť cache a vráti tieto informácie späť volajúcemu.

Počas inštalácie klienta je možné v príkaze **mksecldap** pridať ďalšie voľby ladenia, napríklad nastavenie počtu vlákien používaných procesom typu démon, veľkosti položky v cache pamäti a časový limit pre ukončenie platnosti cache pamäte. Tieto voľby sú len pre skúsených užívateľov. Pre väčšinu prostredí postačia predvolené hodnoty.

V záverečných krokoch nastavenia klienta príkaz **mksecldap** spustí démon na strane klienta a pridá položku do súboru /etc/inittab, aby sa démon spustil pri každom opätovnom zavedení. Úspešnosť nastavenia môžete overiť skontrolovaním procesu démona **secldapcIntd** zadaním príkazu **ls-secldapclntd**. Za predpokladu, že server bezpečnostných informácií LDAP je nainštalovaný a spustený, bude démon po úspešnej inštalácii spustený.

Server musí byť nastavený skôr než klient. Nastavenie klienta je závislé od migrovaných údajov, ktoré sú na serveri. Postupujte podľa týchto krokov pre nastavenie klienta:

1. Nainštalujte klientsku sadu súborov IBM Tivoli Directory Server v operačnom systéme AIX.
   - V aplikácii IBM Tivoli Directory Server 5.2 nainštalujte sadu súborov ldap.client.
   - V aplikácii IBM Tivoli Directory Server 7.1 nainštalujte sadu súborov idsldap.
2. Pre konfiguráciu klienta LDAP spustite nasledujúci príkaz:

   ```
   # mksecldap -c -h server1.ibm.com -a cn=admindn -p adminpwd -d cn=basedn
   ```

   Nahraďte vyššie uvedené hodnoty tak, ako je to vhodné pre vaše prostredie.

Súvisiace informácie:

Príkaz mksecldap

Príkaz secldapclntd

Povolenie klienta pre sieťové skupiny LDAP:

Sieťové skupiny môžete používať ako súčasť NIS-LDAP (metóda rozlíšenie názvu).

Pri zapínaní klienta pre sieťové skupiny LDAP postupujte nasledovne:

1. Nainštalujte a nastavte správu skupín užívateľov na báze LDAP tak, ako je to podrobne uvedené v “Nastavenie klienta LDAP” na strane 148.

   Ak nastavenie sieťovej skupiny nie je dokončené, každý užívateľ definovaný pomocou LDAP bude vypísaný systémom. Napríklad, ak *nguser* je užívateľ sieťovej skupiny patriaci sieťovej skupine *mygroup*, ktorá je už definovaná na LDAP serveri, lsuser -R LDAP nguser vypíše užívateľa.
2. Pre povolenie funkcie netgroup musí definícia modulu pre LDAP v súbore /usr/lib/security/methods.cfg zahrnúť atribút volieb s hodnotou netgroup. Upravte súbor /usr/lib/security/methods.cfg a pridajte riadok options = netgroup do stanzy LDAP. Týmto sa zavádzací modul LDAP označí ako zavádzací modul podporujúci sieťovú skupinu (netgroup). Napríklad:

   ```
   LDAP:
         program = /usr/lib/security/LDAP
         program_64 =/usr/lib/security/LDAP64
         options = netgroup
   ```

   Teraz príkazy lsuser -R LDAP nguser alebo lsuser nguser alebo lsuser -R LDAP -a ALL nevypíšu žiadnych užívateľov. LDAP je teraz považované za databázu len pre sieťovú skupinu z tohto klienta a žiadne sieťové skupiny ešte nemajú povolený prístup k tomuto klientovi.
3. Upravte súbor /etc/passwd a pridajte riadok pre sieťovú skupinu, ktorá by mala mať prístup do systému. Napríklad, ak *mygroup* je sieťová skupina na LDAP serveri, ktorý obsahuje požadovaného užívateľa, pripojte nasledovné:

   ```
   +@mygroup
   ```
4. Upravte súbor /etc/group a pripojte riadok +: pre povolenie hľadaní NIS pre skupiny:

   ```
   +:
   ```

Spustenie príkazu lsuser nguser teraz vráti užívateľa, pretože *nguser* je v sieťovej skupine *mygroup*.

Príkaz lsuser -R LDAP nguser nenájde užívateľa, ale príkaz lsuser -R compat nguser ho nájde, pretože užívateľ je teraz považovaný za užívateľa **compat**.

5. Aby sa užívatelia zo sieťových skupín mohli autentifikovať v systéme, autentifikačný mechanizmus systému AIX musí vedieť, ktorú metódu má použiť. Ak štandardná stanza v súbore /etc/security/user obsahuje SYSTEM = compat, tak všetci užívatelia sieťovej skupiny v sieťovej skupine pridanej do súboru /etc/passwd sa môžu autentifikovať. Ďalšou voľbou by bolo individuálne nakonfigurovať užívateľov tak, že sa manuálne pridajú stanzy do súboru /etc/security/user pre požadovaných užívateľov. Vzorová stanza pre *nguser* je:

   ```
   nguser:
           SYSTEM = compat
           registry = compat
   ```

   Užívatelia sieťovej skupiny v povolených sieťových skupinách sa môžu teraz autentifikovať systému.

   Povolenie funkcie sieťovej skupiny aktivuje aj nasledovné podmienky:

   - Užívatelia definovaní v súbore /etc/security/user ako členovia registra LDAP (majúci registry=LDAP a SYSTEM="LDAP") sa nemôžu autentifikovať ako užívatelia LDAP. Títo užívatelia sú teraz užívateľmi **nis_ldap** a vyžadujú natívne členstvo sieťovej skupiny NIS
   - Význam registra compat je rozšírený tak, že zahŕňa moduly, ktoré používajú sieťovú skupinu. Napríklad, ak modul LDAP podporuje sieťovú skupinu, compat obsahuje súbory, NIS a registre LDAP. Užívatelia získaní z tých modulov majú hodnotu registra compat.

Súvisiace informácie

- Dokument exports File for NFS
- Dokument .rhosts File Format for TCP/IP
- Dokument hosts.equiv File Format for TCP/IP

Podporované servery LDAP:

Správa užívateľov a skupín založená na AIX LDAP podporuje adresárové servery IBM Tivoli, servery iné ako IBM so schémou kompatibilnou s RFC 2307 a aktívne adresárové servery Microsoft.

IBM Tivoli Directory Server

Odporúča sa, aby ste správu užívateľov a skupín v systéme AIX nakonfigurovali prostredníctvom serverov IBM Tivoli Directory Server. Bližšie informácie o konfigurácii serverov IBM Tivoli Directory Server na správu užívateľov a skupín nájdete v téme Konfigurácia servera bezpečnostných informácií IBM Tivoli Directory Server.

Iné ako adresárové servery IBM

AIX podporuje rôzne adresárové servery, ktorých užívatelia a skupiny sú definované pomocou schémy RFC 2307. Keď je systém AIX nakonfigurovaný ako klient LDAP týchto serverov, používa tieto servery rovnakým spôsobom ako server IBM Tivoli Directory Server so schémou RFC 2037. Tieto servery musia podporovať protokol LDAP verzie 3.

Keďže schéma RFC 2307 definuje len podmnožinu atribútov užívateľov a skupín, ktoré môže systém AIX používať, niektoré funkcie správy užívateľov a skupín systému AIX by nebolo možné vykonávať, keby systém AIX nebol nakonfigurovaný na používanie takéhoto servera LDAP (napríklad vynútenie vynulovania hesla užívateľa, história hesiel, limit množstva prostriedkov na užívateľa, riadenie prihlasovania do určitých systémov cez atribúty AIX hostsallowedlogin a hostsdeniedlogin, schopnosti a pod.).

AIX nepodporuje adresárové servery nekompatibilné so schémou RFC 2307. SystémAIX však možno prinútiť spolupracovať s takými servermi, ktoré nie sú kompatibilné so schémou RFC 2307, ale ktorých užívatelia a skupiny sú definované so všetkými vyžadovanými atribútmi systému UNIX. Minimálna množina atribútov užívateľov a skupín vyžadovaná systémom AIX je definovaná v RFC 2307. Podpora takýchto adresárových serverov vyžaduje manuálnu konfiguráciu. AIX poskytuje pre tento účel mechanizmus mapovania schém. Bližšie informácie o formáte súborov

schém a využití súborov schém nájdete v dokumente Formát súboru mapovania atribútov LDAP.

Microsoft Active Directory

AIX podporuje Microsoft Active Directory (AD) ako server LDAP pre správu užívateľov a skupín. Server AD musí mať nainštalovanú schému podporujúcu systém UNIX. Schéma podpory systému UNIX pre AD pochádza z balíka Microsoft Service For UNIX (SFU). Každá verzia SFU má trochu iné definície schém užívateľov a skupín ako jej predchodcovia. AIX podporuje AD spustený na Windows 2000 a 2003 so schémou SFU verzie 3.0 a 3.5 a AD spustený na Windows 2003 R2 so svojou zabudovanou schémou UNIX.

Vzhľadom ba rozdiel v správe užívateľov a skupín medzi systémami UNIX a systémami Windows, nie všetky príkazy systému AIX môžu fungovať na užívateľov LDAP, ak server je AD. K príkazom, ktoré nefungujú, patria **mkuser** a **mkgroup**. Väčšina príkazov pre správu užívateľov a skupín však funguje, v závislosti od prístupových práv udelených identite, s ktorou sa systém AIX viaže na AD. K týmto príkazom patria **lsuser**, **chuser**, **rmuser**, **lsgroup**, **chgroup**, **rmgroup**, **id**, **groups**, **passwd** a **chpasswd**.

AIX podporuje dva mechanizmy autentifikácie užívateľov v porovnaní so servermi Windows: autentifikáciu LDAP a autentifikáciu Kerberos. S každým mechanizmom systém AIX podporuje identifikáciu užívateľov cez protokol LDAP v porovnaní s AD, bez akýchkoľvek požiadaviek na zodpovedajúce konto užívateľa na systéme AIX.

Konfigurácia operačného systému AIX na spoluprácu s Active Directory prostredníctvom LDAP:

AIX podporuje Microsoft Active Directory (AD) ako server LDAP pre správu užívateľov a skupín. Je potrebné, aby mal AD server nainštalovanú schému podporujúcu UNIX.

Administrátor môže pomocou príkazu **mksecldap** nakonfigurovať systém AIX na serveri Active Directory rovnakým spôsobom ako server IBM Tivoli Directory Server. Príkaz **mksecldap** v záujme zjednodušenia procesu skryje všetky podrobnosti týkajúce sa konfigurácie. Skôr ako spustíte príkaz **mksecldap** na konfiguráciu systému AIX na AD serveri:

1. AD server musí mať nainštalovanú schému podpory pre UNIX.
2. AD server musí obsahovať užívateľov, ktorí sú povolené pre UNIX.

Viac informácií o inštalovaní schémy UNIX na AD a o povolení AD užívateľov s podporou UNIX si prečítajte v súvisiacej dokumentácii Microsoft.

AD schéma často obsahuje viacero definícií atribútov pre jeden atribút UNIX (napríklad existuje viacero užívateľských hesiel a definícií členov skupiny). Hoci AIX väčšinu z nich podporuje, mali by ste veľmi dôkladne zvážiť a naplánovať, ktoré definície sa rozhodnete používať. Odporúča sa, aby v záujme vyhnutia sa konfliktom systémy AIX a systémy iné ako AIX, ktoré spoločne zdieľajú AD, používali rovnaké definície.

Výber atribútu hesla Active Directory:

Systém AIX podporuje dva mechanizmy autentifikácie, **unix_auth** a l**dap_auth**.

S **unix_auth** musí byť heslo v Microsoft Active Directory (AD) v šifrovanom formáte. Počas autentifikácie sa šifrované heslo získa z AD a porovná sa so šifrovaným formátom hesla, ktoré zadal užívateľ. Keď sa zhodujú, autentifikácia je úspešná. V režime **ldap_auth** systém AIX autentifikuje užívateľa pomocou LDAP operácie, ktorou vytvorí väzbu na server s identitou užívateľa a zadaným heslom. Ak je operácia vytvorenia väzby úspešná, užívateľ bol autentifikovaný. AD podporuje viacero atribútov užívateľských hesiel. Iný režim autentifikácie v systéme AIX vyžaduje iný atribút užívateľského hesla v AD.

unix_auth mode

V režime **unix_auth** môžu byť použité tieto atribúty hesiel AD:

- **userPassword**
- **unixUserPassword**

- **msSFU30Password**

Správa hesiel v systéme AIX môže byť náročná kvôli viacerým atribútom hesiel v AD. Môže byť mätúce, ktoré atribúty správy hesiel majú použiť klienti systému UNIX. Schopnosť mapovania atribútov v AIX LDAP vám dovoľuje prispôsobiť si správu hesiel podľa vašich potrieb.

Štandardne systém AIX používa atribút **msSFU30Password** pre AD spustený na systémoch Windows 2000 2003, a atribút **userPassword** na systéme Windows 2003 R2. Ak je použité iné heslo, musíte modifikovať súbor /etc/security/ldap/sfu30user.map (alebo súbor /etc/security/ldap/sfur2user.map, ak je AD spustený na systéme Windows 2003 R2). Nájdite riadok začínajúci slovom **spassword** a tretie pole v riadku zmeňte na požadovaný názov hesla AD. Bližšie informácie nájdete v téme Formát mapovacieho súboru atribútov LDAP. Po zmene spustite príkaz **mksecldap** na konfiguráciu klienta AIX LDAP. Ak je klient AIX LDAP už nakonfigurovaný, pomocou príkazu **restart-secldapclntd** reštartujte démona **secldapclntd**, aby sa zmena uplatnila.

V režime **unix_auth** nemusí byť heslo v systéme Windows a v systéme UNIXzosynchronizované, následkom čoho bude pre každý systém platné iné heslo. Stane sa tak v prípade, keď zmeníte heslo z AIX na Windows, pretože systém Windows používa atribút hesla **uncodepwd**. Príkaz AIX **passwd** môže resetovať heslo UNIX, aby bolo rovnaké ako heslo Windows, ale AIX nepodporuje automatickú zmenu hesla Windows, keď zmeníte vaše heslo UNIX z AIX.

ldap_auth mode

Active Directory má tiež atribút hesla **unicodepwd**. Tento atribút hesla používajú systémy Windows na autentifikáciu užívateľov Windows. V operácii vytvorenia väzby na AD sa musí použiť heslo **unicodePwd**. Pre operáciu vytvorenia väzby nefunguje žiadne heslo spomenuté pre režim **unix_auth**. Ak je v príkazovom riadku zadaná voľba **ldap_auth**, príkaz **mksecldap** mapuje atribút hesla na atribút AD **unicodePwd** pri konfigurácii klienta bez nutnosti manuálneho kroku.

Namapovanie AIX hesiel s atribútom **unicodePwd** umožňuje užívateľom definovaným v AD prihlásiť sa do systému Windows aj AIX pomocou rovnakého hesla. Resetovanie hesla buď zo systému AIX alebo Windows je účinné pre oba systémy - AIX aj Windows.

Výber atribútu člena skupiny Active Directory:

Služba Microsoft pre systém UNIX definuje atribúty členov skupiny **memberUid**, **msSFU30MemberUid** a **msSFU30PosixMember**.

Atribúty **memberUid** a **msSFU30MemeberUid** akceptujú názvy kont užívateľov, no **msSFU30PosixMember** akceptuje iba úplný DN. Napríklad pre užívateľské konto *foo* (s priezviskom *bar*), definované v AD:

- **memberUid: foo**
- **msSFU30MemberUid: foo**
- **msSFU30PosixMember: CN=foo bar,CN=Users,DC=austin,DC=ibm,DC=com**

Operačný systém AIX podporuje všetky tieto atribúty. Poraďte sa s administrátorom Active Directory, ktorý atribút máte použiť. Príkaz **mksecldap** štandardne nakonfiguruje operačný systém AIX tak, aby používal atribút **msSFU30PosixMember** pre server Active Directory v systéme Windows 2000 a 2003, a atribút **uidMember** pre server Active Directory v systéme Windows 2003 R2. Tento výber je spôsobený správaním AD, pretože AD vyberie tento atribút pri pridávaní užívateľa do skupiny zo systému Windows. Vaša podniková stratégia môže vyžadovať použitie neštandardného atribútu členov skupiny, aby bolo podporovaných viacero platforiem.

Ak musíte použiť iný atribút členov skupiny, môžete zmeniť mapovanie upravením mapovacieho súboru skupiny. Mapovací súbor skupiny pre AD je /etc/security/ldap/sfu30group.map pre systém Windows 2000 a 2003, a /etc/security/ldap/sfur2group.map pre systém Windows 2003 R2. Nájdite riadok začínajúci slovom **users** a tretie pole nahraďte požadovaným názvom atribútu pre členov skupiny. Bližšie informácie nájdete v téme Formát

mapovacieho súboru atribútov LDAP. Po vykonaní zmeny zadaním príkazu **mksecldap** nakonfigurujte klienta LDAP systému AIX alebo, ak je klient systému AIX už nakonfigurovaný, zadaním príkazu **restart-secldapclntd** reštartujte démona **secldapclntd**, aby sa táto zmena prejavila.

Viacero organizačných jednotiek:

Váš AD server môže mať zadefinovaných viacero organizačných jednotiek, pričom každá z nich obsahuje množinu užívateľov.

Väčšina užívateľov Windows Active Directory je definovaná v podstrome **cn=users,...**, no niektorí môžu byť definovaní inde. V prípade takýchto serverov Active Directory môžete použiť funkciu viacerých základných jedinečných názvov systému AIX. Viac informácií nájdete v téme Podpora viacerých základných DN.

Autentifikácia cez Kerberos pre servery Windows:

Okrem autentifikačných mechanizmov LDAP, operačný systém AIX podporuje aj autentifikáciu užívateľov prostredníctvom protokolu Kerberos pre servery Windows.

Operačný systém AIX podporuje autentifikáciu Kerberos pre identifikáciu Windows KDC a LDAP v adresári Windows Active Directory vytvorením zloženého zavádzacieho modulu KRB5ALDAP. Keďže sa identifikačné informácie užívateľov preberajú z adresára Microsoft Active Directory, v operačnom systéme AIX nemusíte vytvárať príslušné užívateľské kontá.

Správa užívateľov LDAP:

Užívateľov a skupiny môžete spravovať na serveri bezpečnostných informácií LDAP z ľubovoľného klienta LDAP s použitím príkazov vysokej úrovne.

Príznak **-R** pridaný k väčšine vysoko úrovňových príkazov môže spracovať užívateľov a skupiny pomocou LDAP ako aj ostatných zavádzacích modulov autentifikácie, napríklad DCE, NIS a KRB5. Bližšie informácie týkajúce sa použitia príznaku **-R** nájdete v každom príkaze správy užívateľov alebo skupín.

Ak chcete povoliť užívateľov autentifikáciu cez LDAP, spustite príkaz **chuser** na zmenu hodnoty užívateľského atribútu **SYSTEM** na LDAP. Po nastavení hodnoty atribútu **SYSTEM** podľa definovanej syntaxe môže byť užívateľ autentifikovaný cez viac ako jeden zavádzací modul (napríklad compat a LDAP). Ďalšie informácie o nastavovaní užívateľských metód autentifikácie nájdete v časti “Autentifikácia užívateľov” na strane 68 a syntaxe atribútov **SYSTEM** definovanej v súbore /etc/security/user.

Užívateľ sa môže stať užívateľom LDAP v čase nastavenia klienta spustením príkazu **mksecldap** s príznakom **-u** v niektorej z nasledujúcich foriem:

1. Spustite príkaz:

   ```
   mksecldap -c -u user1,user2,...
   ```

   kde *user1,user2,...* je zoznam užívateľov. Užívatelia na tomto zozname môžu byť lokálnymi alebo vzdialene cez LDAP definovanými užívateľmi. Atribút **SYSTEM** je nastavený na hodnotu LDAP v každej predchádzajúcej užívateľskej sekcii v súbore /etc/security/user. Autentifikácia takýchto užívateľov je možná len prostredníctvom protokolu LDAP. Užívatelia v tomto zozname musia existovať na serveri LDAP security information server, v opačnom prípade nie je možné ich pripojenie z tohto hostiteľa. Spustite príkaz **chuser** na modifikáciu atribútu **SYSTEM** a umožníte autentifikáciu pomocou viacerých metód (napríklad lokálnej a LDAP).
2. Spustite

   ```
   mksecldap -c -u ALL
   ```

   Tento príkaz nastaví atribút **SYSTEM** na hodnotu LDAP v každej užívateľskej sekcii v súbore /etc/security/user pre všetkých lokálne definovaných užívateľov. Autentifikácia týchto užívateľov je možná len prostredníctvom protokolu LDAP. Lokálne definovaní užívatelia musia existovať na serveri LDAP security information server, v

opačnom prípade nie je možné ich pripojenie z tohto hostiteľa. Užívateľ definovaný na serveri LDAP, ktorý však nie je definovaný lokálne, sa nemôže prihlásiť z tohto hostiteľa. Ak chcete povoliť prihlásenie vzdialeného cez LDAP definovaného užívateľa z tohto hostiteľa, spustite príkaz **chuser** na nastavenie atribútu **SYSTEM** na LDAP pre daného užívateľa.

Ďalšia možnosť je aktivovať pre všetkých užívateľov LDAP bez ohľadu na to, či sú definovaní lokálne, autentifikáciu pomocou protokolu LDAP na lokálnom hostiteľovi, a to modifikáciou predvolenej sekcie súboru /etc/security/user na hodnotu LDAP. Všetci užívatelia, ktorí pre svoj atribút **SYSTEM** nemajú definovanú hodnotu, musia postupovať podľa definície v predvolenej sekcii. Napríklad, ak štandardná stanza má "SYSTEM = "compat"", jej zmena na "SYSTEM = "compat OR LDAP"" umožní autentifikáciu týchto užívateľov buď prostredníctvom AIX alebo LDAP. Zmena štandardnej stanzy na "SYSTEM = "LDAP"" umožní týmto užívateľom autentifikovať sa výlučne prostredníctvom LDAP. Tí užívatelia, ktorí majú definovanú hodnotu atribútu **SYSTEM**, nie sú ovplyvnení predvolenou sekciou.

Podpora viacerých základných DN:

Systém AIX podporuje viacero základných DN. V súbore /etc/security/ldap/ldap.cfg môžete zadať do 10 základných DN pre jednu entitu.

Základné DN majú prioritu danú poradím, v ktorom sú uvedené v súbore /etc/security/ldap/ldap.cfg. Príkazy AIX sa v prípade viacerých základných DN vykonávajú podľa priority, ktorú má základné DN, a prebieha to takto:

- Operácia dotazu (volaná napríklad príkazom **lsuser**), sa vykonáva na základných DN podľa ich priority, kým sa nenájde vyhovujúce konto, alebo kým sa nevráti zlyhanie v prípade, že boli prehľadané všetky základné DN a zhoda sa nenašla. Pri dotaze na všetky základné DN budú vrátené všetky kontá všetkých základných DN.
- Operácia modifikácie (volaná napríklad príkazom **chuser**) sa vykoná na prvom vyhovujúcom konte.
- Operácia vymazania (volaná napríklad príkazom **rmuser**) sa vykoná na prvom vyhovujúcom konte.
- Operácia vytvorenia (volaná napríklad príkazom **mkuser**) sa vykoná iba na prvom základnom DN. AIX nepodporuje vytváranie kont pre iné základné DN.

Je zodpovednosťou administrátora adresárového servera uchovávať databázu kont v bezkonfliktnom stave. Ak existuje viacero definícií toho istého konta, každá v inom podstrome, pre systém AIX je viditeľné iba prvé konto. Operácia vyhľadávania vráti iba prvé vyhovujúce konto. Podobne aj operácia modifikácie alebo vymazania sa vykoná iba na prvom vyhovujúcom konte.

Príkaz **mksecldap**, keď sa použije na konfiguráciu klienta LDAP, nájde základné DN pre každú entitu a uloží ho do súboru /etc/security/ldap/ldap.cfg. Keď pre jednu entitu na LDAP serveri existuje viacero základných DN, príkaz **mksecldap** náhodne vyberie a použije ľubovoľné z nich. Ak chcete, aby systém AIX fungoval s viacerými základnými DN, musíte po úspešnom dokončení príkazu **mksecldap** upraviť súbor /etc/security/ldap/ldap.cfg. Nájdite príslušnú definíciu základného DN a pridajte ďalšie potrebné základné DN. AIX podporuje maximálne 10 základných DN pre každú entitu, akékoľvek ďalšie základné DN budú ignorované.

AIX tiež podporuje užívateľom definovaný filter a rozsah vyhľadávania pre každé základné DN. Základné DN môže mať svoj vlastný filter a rozsah, ktoré môžu byť iné ako u rovesníckych základných DN. Filtre sa dajú použiť na definovanie množiny kont, ktoré budú viditeľné pre systém AIX.

Systém AIX vidí iba kontá, ktoré prejdú filtrom.

Nastavenie SSL na serveri LDAP:

Pri nastavovaní protokolu SSL (Secure Sockets Layer) na serveri LDAP nainštalujte sady súborov LDAP a **GSKit**, ktoré umožňujú podporu šifrovania na serveri. Tieto sady súborov možno nájsť v rozširujúcom balíku AIX.

Ak chcete povoliť podporu SSL pre autentifikáciu servera IBM Directory, dodržte nasledujúci postup.

1. Nainštalujte balík IBM Tivoli Directory Server **GSKit** pre IBM Tivoli Directory Server, verzia 6.2, alebo **GSKitv8** pre IBM Tivoli Directory Server, verzia 6.3, ak ešte nie je nainštalovaný.

2. Vygenerujte súkromný kľúč servera IBM Directory Server a serverový certifikát pomocou príslušného pomocného programu na správu kľúčov GSKit. V prípade servera IBM Tivoli Directory Server, verzia 6.2, použite pomocný program **gsk7ikm**, kým v prípade servera IBM Tivoli Directory Server, verzia 6.3, alebo novšieho, použite nástroj ikeyman. Serverový certifikát môže byť podpísaný komerčnou certifikačnou autoritou (CA), ako napríklad VeriSign, alebo môže byť samopodpísaný nástrojom na správu kľúčov GSKit. Verejný certifikát (alebo samopodpísaný certifikát) certifikačnej autority sa musí tiež distribuovať do súboru databázy kľúčov klientskej aplikácie.
3. Uložte súbor databázy kľúčov servera a súvisiaci súbor skladu hesiel na server. Zvyčajným umiestnením adresára /usr/ldap/etc je štandardná cesta pre databázu kľúčov.
4. Zadaním nasledujúceho príkazu nakonfigurujte server, pričom **mykey.kdb** predstavuje databázu kľúčov a *keypwd* je heslo pre databázu kľúčov:

   ```
   , mksecldap -s -a cn=admin -p pwd -S rfc2307aix -k /usr/ldap/etc/mykey.kdb -w keypwd,
   ```

Nastavenie SSL na klientovi LDAP:

Ak chcete používať protokol SSL na klientovi LDAP, nainštalujte sady súborov ldap.max_crypto_client a GSKit z disku AIX Expansion Pack.

Ak chcete povoliť podporu SSL pre LDAP, keď ste server povolili pre SSL, dodržte nasledujúci postup.

1. Ak chcete vygenerovať databázu kľúčov na každom klientovi, spustite gsk7ikm.
2. Skopírujte certifikát servera do každého z klientov. Ak SSL servera používa samopodpisový certifikát, certifikát sa musí exportovať ako prvý.
3. Ak chcete naimportovať certifikát servera do databázy kľúčov, na každom klientskom systéme spustite gsk7ikm.
4. Povoľte SSL pre každého klienta:

   ```
   , mksecldap -c -h servername -a adminDN -p pwd -k /usr/ldap/etc/mykey.kdb -p keypwd,
   ```

   Pričom /usr/ldap/etc/mykey.kdb je úplná cesta k databáze kľúčov a *keypwd* je heslo pre kľúč. Ak sa heslo kľúča nezadá z príkazového riadka, použije sa súbor hesiel z rovnakého adresára. Ukrytý súbor musí mať rovnaký názov ako databáza kľúčov s príponou **.sth** (napríklad, mykey.sth).

Riadenie prístupov k hostiteľovi LDAP:

Systém AIX poskytuje ovládanie prístupu (prihlásenia) k hostiteľovi na úrovni užívateľa. Administrátori môžu nakonfigurovať užívateľov LDAP, aby sa prihlasovali so systému AIX pomocou nastavenia atribútu **SYSTEM** na hodnotu LDAP.

Atribút **SYSTEM** sa nachádza v súbore /etc/security/user. Na nastavenie jeho hodnoty možno použiť príkaz **chuser**, a to nasledovným spôsobom:

```
# chuser -R LDAP SYSTEM=LDAP registry=LDAP xy
```

Poznámka: Pri tomto type riadenia nenastavujte predvolený atribút **SYSTEM** na hodnotu LDAP, ktorá umožní všetkým užívateľom LDAP prihlasovanie do systému.

Tu sa atribút LDAP nastaví tak, aby umožňoval užívateľovi *xy* prihlasovanie do tohto systému. Ďalej nastaví databázu Registry na LDAP, čo v procese prihlásenia zabezpečí protokolovanie pokusov o prihlásenie užívateľa *xy* do systému LDAP a umožní tiež vykonávanie úloh správy užívateľov v systéme LDAP.

Administrátor musí spustiť toto nastavenie v každom klientskom systéme, aby bolo možné prihlásenie jednotlivých užívateľov.

Systém AIX umožňuje obmedziť prihlásenie užívateľov LDAP na konkrétne klientske systémy LDAP. Táto funkcia umožňuje centralizovanú správu ovládania prístupov k hostiteľovi. Administrátori majú možnosť určiť dva zoznamy prístupu k hostiteľovi pre užívateľské konto: zoznam povoleného prístupu (allow list) a zoznam nepovoleného prístupu (deny list). Tieto dva atribúty užívateľa sú uložené na serveri LDAP s užívateľským kontom. Užívateľ má povolený

prístup do systémov a sietí špecifikovaných v zozname povoleného prístupu, zatiaľ čo má zamietnutý prístup do systémov a sietí v zozname nepovoleného prístupu. Ak je systém zadaný v oboch zoznamoch, užívateľ má zamietnutý prístup do systému. Existujú dva spôsoby špecifikovania zoznamov prístupových práv pre užívateľa: s príkazom **mkuser**, keď sa vytvára užívateľ, alebo s príkazom **chuser** pre existujúceho užívateľa. Pre účely spätnej kompatibility, ak pre užívateľa neexistuje zoznam povoleného ani nepovoleného prístupu, užívateľ sa môže štandardne prihlásiť do ľubovoľného klientskeho systému LDAP.

Nasledujú príklady nastavenia zoznamov povolených a nepovolených oprávnení pre užívateľov:

```
# mkuser -R LDAP hostsallowedlogin=host1,host2 xy
```

Týmto sa vytvorí užívateľ *foo* a užívateľ *foo* má povolené prihlásiť sa iba na host1 a host2.

```
# mkuser -R LDAP hostsdeniedlogin=host2 xy
```

Týmto sa vytvorí užívateľ *foo* a užívateľ *foo* sa môže prihlásiť na ľubovoľné klientske systémy LDAP, s výnimkou host2.

```
# chuser -R LDAP hostsallowedlogin=192.9.200.1 xy
```

Týmto sa nastaví užívateľ *foo* s oprávnením prihlásiť sa na klientsky systém na adrese 192.9.200.1.

```
# chuser -R LDAP hostsallowedlogin=192.9.200/24 hostsdeniedlogin=192.9.200.1 foo
```

Týmto sa nastaví užívateľ *foo* s oprávnením prihlásiť sa na ľubovoľný klientsky systém v podsieti 192.9.200/24, s výnimkou klientskeho systému na adrese 192.9.200.1.

Ak chcete získať viac informácií, pozrite si príkaz **chuser**.

Komunikácia zabezpečená pomocou SSL:

Podľa toho aký typ autentifikácie sa používa medzi serverom a klientom LDAP, heslá sa budú odosielať buď v šifrovanom formáte (unix_auth) alebo ako čitateľný text (ldap_auth). Na ochranu proti ohrozeniu bezpečnosti pri zasielaní dokonca aj šifrovaných hesiel cez sieť alebo v niektorých prípadoch cez internet použite SSL (Secure Socket Layer). AIX poskytuje balíky pre SSL, ktoré poskytujú bezpečnú komunikáciu medzi adresárovými servermi a klientmi.

Bližšie informácie nájdete v:

- “Nastavenie SSL na serveri LDAP” na strane 154
- “Nastavenie SSL na klientovi LDAP” na strane 155

Používanie režimu LDAPA len na autentifikáciu:

Modul LDAP je plne funkčný modul, ktorý podporuje autentifikáciu aj identifikáciu užívateľa. Modul LDAPA poskytuje len režim na autentifikáciu. Modul LDAPA je ako modul LDAP, ale môžete zadať, aby sa používal len v režime na autentifikáciu.

V režime len na autentifikáciu musí byť modul LDAPA skombinovaný s iným modulom databázy, aby mohol vytvoriť zmiešaný modul, nie samostatný modul. Modul LDAPA vykoná autentifikáciu užívateľa zatiaľ čo druhý modul vykoná identifikáciu. Tento kombinovaný modul sa volá zmiešaný modul. Pre tento zmiešaný modul musíte zadefinovať užívateľov na LDAP serveri aj na databázovom serveri.

Ak používate modul LDAPA, skupinové informácie budú prichádzať z databázového servera. Napríklad v prípade súborov LDAPA budú skupinové informácie prichádzať z lokálneho súboru /etc/group. Ak niektorí z vašich užívateľov LDAP patria len do skupín LDAP, skôr ako nakonfigurujete modul súborov LDAPA musíte na databázovom serveri vytvoriť príslušné skupiny LDAP. Vytvorením tejto príslušnej skupiny sa môžete vyhnúť situácii, v ktorej užívateľ súborov LDAPA nemôže vyriešiť svoje nastavenie skupiny, pretože dané nastavenie skupiny na databázovom serveri neexistuje.

Poznámka: Modul LDAPA nepodporuje vytváranie a odstraňovanie užívateľov. Ak chcete vytvoriť užívateľa súborov LDAPA, administrátor systému musí pomocou modulu LDAP vytvoriť užívateľa LDAP a následne vytvoriť rovnakého užívateľa lokálne. Z tohto užívateľa následne urobte užívateľa súborov LDAPA tak, že príkazom **chuser** nastavíte SYSTEM a register užívateľa na LDAPAfiles.

Ak chcete konfigurovať LDAP v režime len na autentifikáciu pomocou modulu LDAPA, použite príkaz **mksecldap** s voľbou -i <*databaseModule*>. Tento príkaz vytvorí modul LDAPA so sadou options = authonly a zmiešaný zavádzací modul <*databaseModule*> LDAP.

Ak chcete napríklad konfigurovať LDAP v režime len na autentifikáciu a použiť súbory pre modul databázy, použite tento príklad:

```
mksecldap -c –h <ldap server> -a <binddn> -p <bind password> -i files
```

Súbor /usr/lib/security/methods.cfg je aktualizovaný s:

```
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
```

Nastavenie options = authonly v odseku LDAPA indikuje, aby sa modul LDAPA nastavil na režim len na autentifikáciu. Odsek LDAPAfiles definuje zmiešaný zavádzací modul.

Modul LDAP je ponechaný na vyriešenie neužívateľských/skupinových údajov, napríklad RBAC. Modul LDAP možno napriek tomu použiť ako samostatný modul autentifikácie nezávislý od modulu LDAPA.

Súvisiace informácie:

Príkaz mksecldap

Atribúty podporované modulom LDAPA:

Modul LDAPA v režime len na autentifikáciu podporuje obmedzený počet atribútov politiky hesiel systému AIX. Zvyšné atribúty systému AIX poskytuje databázový modul.

Modul LDAPA len na autentifikáciu podporuje nasledujúce atribúty:

- maxage
- minage
- minlen
- lastupdate
- príznaky
- 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

Nie všetky LDAP servery podporujú tieto atribúty. Keď LDAP server nepodporuje všetky vypísané atribúty, podporované atribúty sú len atribúty bežné v oboch zoznamoch a v mapovacom súbore užívateľ - atribút. Mapovací súbor je v adresári /etc/security/ldap.

V prípade servera v súlade so štandardom RFC2307 bez podpory schém AIX sú podporované nasledujúce atribúty systému AIX:

- maxage
- minage
- lastupdate
- pwdwarntime
- lastupdate

Väzba na Kerberos:

Okrem jednoduchej väzby s použitím DN väzby a hesla väzby démon **secldapclntd** podporuje aj väzbu s použitím oprávnení Kerberos V.

Kľúče principála väzby sú uložené v súbore keytab a musia byť sprístupnené démonu **secldapclntd**, aby mohol používať väzbu na Kerberos. Ak je väzba na Kerberos povolená, démon **secldapclntd** uskutoční autentifikáciu Kerberos pre LDAP server s použitím názvu principála a keytab zadaného v súbore konfigurácie klienta /etc/security/ldap/ldap.cfg. Použitie väzby na Kerberos spôsobí, že démon **secldapclntd** bude ignorovať DN väzby a heslo väzby zadané v súbore /etc/security/ldap/ldap.cfg.

Keď bude autentifikácia Kerberos úspešná, démon **secldapclntd** uloží oprávnenia väzby do adresára /etc/security/ldap/krb5cc_secldapclntd. Uložené povoľovacie údaje sa použijú na neskoršie opakované vytvorenie väzby. Ak sú povoľovacie údaje v čase, keď sa démon **secldapclntd** pokúša o opakované vytvorenie väzby na server LDAP, staršie ako jednu hodinu, potom démon **secldapclntd** bude opätovne inicializovať obnovu povoľovacích údajov.

Ak si želáte nakonfigurovať klientsky systém LDAP na používanie väzby na systém Kerberos, musíte tohto klienta konfigurovať pomocou príkazu **mksecldap** a s použitím charakteristického názvu a hesla väzby. Ak konfigurácia prebehne úspešne, zapíšte do súboru /etc/security/ldap/ldap.cfg správne hodnoty atribútov vzťahujúcich sa na systém Kerberos. Démon **secldapclntd** použije pri reštarte väzbu na Kerberos. Po úspešnej konfigurácii sa už charakteristický názov a heslo väzby nepoužijú. Zo súboru /etc/security/ldap/ldap.cfg ich možno bezpečne odstrániť alebo znefunkčniť návestiami komentára.

Vytvorenie principála Kerberos:

Aby ste podporili väzbu na Kerberos, musíte v distribučnom centre kľúčov (KDC) vytvoriť najmenej dva principály, ktoré bude používať server a klient IDS. Prvým principálom je principál servera LDAP a druhým je principál, ktorý používajú klientske systémy pri vytváraní väzieb na tento server.

Každý z kľúčov principála musí byť umiestnený v súbore tabuľky kľúčov, aby mohli byť použité na spustenie procesu servera alebo procesu démona klienta.

Nasledujúci príklad je založený na službe IBM Network Authentication Service. Ak inštalujete softvér Kerberos z iných zdrojov, konkrétne príkazy sa môžu líšiť od tých, ktoré uvádzame tu.

- Spustite nástroj kadmin na serveri KDC ako užívateľ s oprávneniami typu root.

```
#/usr/krb5/sbin/kadmin.local
kadmin.local:
```

- Vytvorte principál ldap/*serverhostname* pre LDAP server. Hostiteľ *serverhostname* je plne kvalifikovaný hostiteľ DNS, ktorý spustí server LDAP.

```
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:
```

- Vytvorte tabuľku kľúčov pre vytvorený principál servera. Tento kľúč použije server LDAP počas spúšťania servera. Vytvorenie tabuľky kľúčov s názvom slapd_krb5.keytab:

```
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:
```

- Vytvorte principál s názvom ldapadmin pre administrátora IDS.

```
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:
```

- Vytvorte tabuľku kľúčov pre principál väzby kdapadmin.keytab. Tento kľúč môže používať klientsky démon **secldapclntd**.

```
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
```

- Vytvorte principála s názvom ldapproxy, aby klienti mohli vytvoriť väzby na LDAP server.

```
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:
```

- Vytvorte keytab s názvom *ldapproxy.keytab* pre principála väzby **ldapproxy**. Tento kľúč môže používať klientsky démon **secldapclntd**.

```
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:
```

Povolenie väzby na Kerberos servera IDS:

Nasledujúca procedúra zapína server IDS pre vytvorenie väzby Kerberos.

Nasledujúca ukážka ilustruje konfiguráciu servera IDS pre väzbu na Kerberos.

Tento príklad bol testovaný s použitím IDS v5.1:

1. Nainštalujte sadu súborov krb5.client.
2. Presvedčte sa, či existuje súbor /etc/krb5/krb5.conf a či je správne nakonfigurovaný. Ak je ho treba konfigurovať, môžete spustiť príkaz **/usr/sbin/config.krb5**.

```
# 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. Získajte súbor keytab principála ldap:/*serverhostname* a umiestnite ho do adresára /usr/ldap/etc. Napríklad: /usr/ldap/etc/slapd_krb5.keytab.
4. Nastavte oprávnenie tak, aby mal proces servera k súboru prístup.

```
# chown ldap:ldap/usr/ldap/etc/slapd_krb5.keytab
#
```

5. Pre povolenie servera DS pre väzbu na Kerberos upravte súbor /etc/ibmslapd.conf a pripojte nasledujúcu položku:

```
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. Mapujte principála ldapproxy na väzbu DN s názvom cn-proxyuser,cn=aixdata.
   a. Ak položka s charakteristickým názvom väzby na IDS serveri existuje, vytvorte súbor s názvom ldapproxy.ldif s nasledovným obsahom:

      ```
      dn: cn=proxyuser,cn=aixdata
      changetype: modify
      add: objectclass
      objectclass: ibm-securityidentities
      -
      add:altsecurityidentities
      altsecurityidentities: Kerberos:ldapproxy@ud3a.austin.ibm.com
      ```

      ALEBO

   b. Ak položka s charakteristickým názvom väzby na IDS server ešte pridaná nebola, vytvorte súbor s názvom proxyuser.ldif s nasledovným obsahom:

      Poznámka: Budete musieť nahradiť *proxyuserpwd* vaším heslom.

      ```
      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
      ```

      Pomocou príkazu **ldapmodify** pridajte vytvorenú položku s charakteristickým názvom väzby na server IDS.

      ```
      # ldapmodify -D cn-admin -w adminPwd -f /tmp/proxyuser.ldif modifying entry cn=proxyuser,cn=mytest
      #
      ```

7. Reštartujte IDS server.

Povolenie väzby na Kerberos klienta LDAP AIX:

Môžete nakonfigurovať systém klienta LDAP AIX na použitie Kerberos vo svojej úvodnej väzbe na server LDAP.

Server IDS musí byť týmto spôsobom nakonfigurovaný na hostiteľa servera, aby bol klientom sám sebe.

Tento príklad bol testovaný s použitím IDS v 5.1:

1. Nainštalujte sadu súborov krb5.client.
2. Presvedčte sa, či existuje súbor /etc/krb.conf a či je správne nakonfigurovaný. Ak nie je správne nakonfigurovaný, môžete spustiť príkaz **/usr/sbin/config.krb5** a nakonfigurovať ho.
3. Z principála väzby si vezmite súbor keytab a umiestnite ho do adresára /etc/security/ldap.
4. Oprávnenie nastavte na 600.
5. Pomocou príkazu **mksecldap** a s použitím charakteristického názvu a hesla väzby nakonfigurujte klienta. Uistite sa, že príkazy AIX na používateľoch LDAP fungujú.
6. Úpravami v súbore /etc/security/ldap/ldap.cfg nastavte atribúty týkajúce sa systému Kerberos. V nasledujúcom príklade je **ldapproxy** principál väzby a súbor tabuľky kľúčov je **ldapproxy.keytab**. Ak chcete oprávnenia administrátora servera IDS, vymeňte *ldapproxy* za *ldapadmin* a *ldapproxy.keytab* vymeňte za *ldapadmin.keytab*.

   ```
   useKRB5:yes
   krbprincipal:ldapproxy
   krbkeypath:/etc/security/ldap/ldapproxy.keytab
   krbcmddir:/usr/krb5/bin/
   ```

Teraz môže byť DN väzby a heslo väzby odstránené zo súboru ldap.cfg alebo označené ako komentár, pretože démon **secldapclntd** teraz používa väzbu Kerberos.

7. Reštartujte démona **secldapclntd**.
8. Súbor /etc/security/ldap/ldap.cfg možno teraz kopírovať do ďalších klientskych systémov.

Auditovanie servera bezpečnostných informácií LDAP:

SecureWay Directory Verzia 3.2 (a novší) poskytuje predvolenú funkciu protokolovania auditu servera. Po povolení bude tento predvolený modul plug-in auditu protokolovať aktivity servera LDAP do protokolového súboru. Bližšie informácie o tomto predvolenom module plug-in nájdete v dokumentácii k adresáru LDAP v príručke *Packaging Guide for LPP Installation*.

Funkcia auditovania servera bezpečnostných informácií LDAP, ktorá sa poskytuje s operačným systémom AIX, sa nazýva *modul plug-in bezpečnostného auditu LDAP*. Existuje nezávisle od štandardnej auditovacej služby systému SecureWay Directory, takže je možné zapnúť jeden alebo obidva z týchto auditovacích subsystémov. Modul plug-in auditu v systéme AIX zaznamenáva iba tie udalosti, ktoré upravujú alebo získavajú bezpečnostné informácie systému AIX o serveri LDAP. Funguje v rámci auditovania systému AIX.

V súbore /etc/security/audit/event sú kvôli zahrnutiu LDAP obsiahnuté nasledovné udalosti auditu:

- LDAP_Bind
- LDAP_Unbind
- LDAP_Add
- LDAP_Delete
- LDAP_Modify
- LDAP_Modifydn
- LDAP_Search

Definícia triedy auditu ldapserver sa vytvorí aj v súbore /etc/security/audit/config, ktorý obsahuje všetky vyššie uvedené udalosti.

Na auditovanie servera LDAP security information server pridajte nasledovný riadok do sekcie každého užívateľa v súbore /etc/security/audit/config:

```
ldap = ldapserver
```

Keďže plug-in auditu servera LDAP security information server je implementovaný v rámci auditovania systému AIX, je súčasťou auditovacieho podsystému systému AIX. Auditovanie servera bezpečnostných informácií LDAP môžete povoliť alebo zakázať pomocou systémových príkazov auditovania, ako sú **audit start** a **audit shutdown**. Všetky záznamy auditu sa pridávajú do auditovacieho záznamu systému, ktorý je možné prezerať pomocou príkazu **auditpr**. Viac informácií nájdete v časti “Prehľad auditu” na strane 127.

Príkazy LDAP:

Existuje niekoľko príkazov LDAP.

príkaz lsldap

Príkaz **lsldap** môže byť použitý na zobrazenie entít pomenúvacej služby z nakonfigurovaného LDAP servera. Tieto entity sú aliases, automount, bootparams, ethers, groups, hosts, netgroups, networks, passwd, protocols, rpc a services.

príkaz mksecldap

Príkaz **mksecldap** môže byť použitý na nastavenie adresárových serverov a klientov IBM SecureWay pre bezpečnostnú autentifikáciu a manažment údajov. Tento príkaz sa musí spustiť na serveri a všetkých klientoch.

démon secldapclntd

Démon **secldapclntd** prijíma požiadavky zo zavádzacieho modulu LDAP, postúpi danú požiadavku na server LDAP Security Information Server a odovzdá výsledok zo servera späť do zavádzacieho modulu LDAP.

Príkazy manažmentu LDAP:

Na riadenie LDAP sa používa niekoľko príkazov.

príkaz start-secldapclntd

Príkaz **start-secldapclntd** spustí proces démon **secldapclntd**, ak démon nie je spustený.

príkaz stop-secldapclntd

Príkaz **stop-secldapclntd** ukončí spustenie procesu démon **secldapclntd**.

príkaz restart-secldapclntd

Skript **restart-secldapclntd** ukončí proces démon **secldapclntd**, ak je spustený a potom ho reštartuje. V prípade, že démon **secldapclntd** nie je spustený, spustí ho.

príkaz ls-secldapclntd

Príkaz **ls-secldapclntd** zobrazí stav démona **secldapclntd**.

príkaz flush-secldapclntd

Príkaz **flush-secldapclntd** vymaže cache pamäť pre proces démon **secldapclntd**.

príkaz sectoldif

Príkaz **sectoldif** prečíta užívateľov a skupiny definované lokálne a vytlačí výsledky na štandardný výstup vo formáte **ldif**.

Mapovanie formátu súboru pre atribúty LDAP:

Tieto mapové súbory používa modul /usr/lib/security/LDAP a démon **secldapclntd** na preklad názvov atribútov AIX na názvy atribútov LDAP.

Každá položka v mapovacom súbore predstavuje preklad atribútu. Položka má štyri polia oddelené medzerami:

```
AIX_Attribute_Name AIX_Attribute_Type LDAP_Attribute_Name LDAP_Value_Type
```

Nasledujú popisy týchto polí:

AIX_Attribute_Name
Určuje názov atribútu AIX.

AIX_Attribute_Type
Určuje typ atribútu AIX. Hodnoty sú SEC_CHAR, SEC_INT, SEC_LIST a SEC_BOOL.

LDAP_Attribute_Name
Určuje názov atribútu LDAP.

LDAP_Value_Type
Určuje typ hodnoty LDAP. Hodnoty sú **s** pre jednu hodnotu a **m** pre viac hodnôt.

LDAP a KRB5LDAP v jednom klientovi

Ak je LDAP súčasťou zloženého modulu, napríklad KRB5LDAP, možné sú len operácie čítania, nie operácie zápisu. Avšak pomocou ďalej uvedených zmien konfigurácie v súbore **/usr/lib/security/methods.cfg** môžete LDAP aj zložené zavádzacie moduly, napríklad KRB5LDAP, umiestniť do jedného súboru vykonaním týchto krokov:

1. Nakonfigurujte klienta LDAP a klientov KRB5LDAP ako obyčajne.
2. Súbor **/usr/lib/security/methods.cfg** upravte takto:

```
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. Súbor **/etc/security/user** upravte pre predvolený odsek takto:

```
        SYSTEM = "KRB5LXAP OR LDAP OR compat"
```

   Užívatelia LDAP môžu byť spracovaní ako obyčajne. Nasledujúce príklady znázorňujú spracovanie užívateľov KRB5LDAP:

```
mkuser -R KRB5LXAP <meno_užívateľa>
rmuser -R KRB5LXAP <meno_užívateľa>
lsuser -R KRB5LXAP <meno_užívateľa>
passwd -R KRB5LXAP <meno_užívateľa>
```

EFS - Encrypted File System

Systém EFS umožňuje jednotlivým užívateľom systému zašifrovať svoje údaje v súborovom systéme J2 cez ich individuálne sklady kľúčov.

Každému užívateľovi je priradený kľúč. Kľúče sú uložené v zašifrovanom sklade kľúčov. Keď sa užívateľ úspešne prihlási, jeho kľúče sa načítajú do kernelu a asociujú sa so splnomocneniami procesov. Neskôr, keď proces potrebuje otvoriť súbor chránený cez EFS, sa tieto splnomocnenia otestujú, a keď sa nájde kľúč zodpovedajúci ochrane súboru, proces môže dešifrovať kľúč k súboru a tým aj obsah súboru. Podporovaná je aj správa kľúčov založená na skupinách.

Poznámka: EFS tvorí súčasť celkovej bezpečnostnej stratégie. Je navrhnutý tak, aby fungoval v súčinnosti s ďalšími postupmi a nástrojmi, ktoré sa starajú o počítačovú bezpečnosť.

Použiteľnosť systému EFS

Správa kľúčov systému EFS (Encrypted File System), šifrovanie a dešifrovanie súborov je transparentné pre užívateľov v bežných operáciách.

EFS je súčasťou základného operačného systému AIX. Ak chcete aktivovať EFS, užívateľ root (alebo akýkoľvek iný užívateľ s oprávnením RBAC **aix.security.efs** - bližšie informácie nájdete v časti Roly EFS) musí pomocou príkazu **efsenable** aktivovať EFS a vytvoriť prostredie EFS. Toto je jednorazová aktivácia v systéme. Keď je EFS povolené, pri prihlásení užívateľa je v pozadí vytvorený kľúč a sklad kľúčov a zabezpečený alebo zašifrovaný prihlasovacím heslom užívateľa. Kľúče užívateľov sú používané na pozadí súborovým systémom J2 na šifrovanie a dešifrovanie súborov EFS. Každý súbor EFS je zabezpečený jedinečným kľúčom súboru a tento kľúč súboru je zabezpečený alebo zašifrovaný kľúčom vlastníka súboru alebo skupiny, v závislosti od povolení súboru.

Štandardne nie je súborový systém J2 povolený pre EFS. Keď je aktivovaný pre EFS, súborový systém J2 transparentne riadi šifrovanie a dešifrovanie v kerneli pre požiadavky na čítanie a zapisovanie. Administračné príkazy užívateľov a skupín (napríklad **mkgroup**, **chuser** a **chgroup**) transparentne spravujú sklady kľúčov užívateľov a skupín.

Nasledujúce EFS príkazy umožňujú užívateľom spravovať svoje kľúče a šifrovanie súborov:

efskeymgr
Riadi a spravuje kľúče

efsmgr Spravuje šifrovanie súborov/adresárov/súborového systému

Užívatelia Encrypted File System

Existujú tri typy užívateľov, ktorí môžu spravovať a používať kľúče EFS:

Úplný alebo obmedzený koreňový prístup:

Prístup užívateľa typu root ku kľúčom môže byť neobmedzený alebo obmedzený. V žiadnom z týchto režimov nemôže užívateľ root vykonať príkaz **su** na užívateľa a získať tak prístup ku skladu kľúčov alebo zašifrovanému súboru užívateľa.

V jednom režime môže koreňový užívateľ resetovať heslo ku skladu kľúčov užívateľa, a môže získať prístup k užívateľovým kľúčom v danom sklade kľúčov. Tento režim poskytuje väčšiu pružnosť pri správe systému.

V inom režime môže koreňový užívateľ resetovať prihlasovacie heslo užívateľa, no nemôže resetovať heslo ku skladu kľúčov užívateľa. Koreňový užívateľ nemôže nahradiť užívateľa (príkazom **su**) a zdediť otvorený sklad kľúčov. Hoci koreňový užívateľ môže vytvárať a vymazávať užívateľov a skupiny spolu s ich priradenými skladmi kľúčov, nemôže získať prístup ku kľúčom v týchto skladoch kľúčov. Tento režim poskytuje vyšší stupeň ochrany voči útokom zo strany zlomyseľného koreňového užívateľa.

Existujú dva režimy na správu a používanie skladov kľúčov: Root Admin a Root Guard. Poskytnutý je tiež administračný kľúč pre EFS.

Administračný kľúč pre EFS umožňuje prístup k heslám do všetkých skladov kľúčov v režime Root Admin. Tento kľúč je umiestnený v špeciálnom sklade kľúčov **efs_admin**. Prístup k tomuto špeciálnemu skladu kľúčov **efs_admin** je poskytnutý iba autorizovaným užívateľom (koreňový užívateľ a bezpečnostná skupina pri inštalácii, alebo autorizácia RBAC **aix.security.efs**).

Keď je sklad kľúčov v režime Root Guard, kľúče nachádzajúce sa v tomto sklade kľúčov sa nedajú získať bez správneho hesla. To poskytuje silné zabezpečenie proti potenciálnemu zlomyseľnému koreňovému užívateľovi, no môže tiež spôsobiť problémy, ak nejaký užívateľ zabudne svoje heslo - neexistuje totiž spôsob, ako regenerovať heslo bez straty kľúčov v sklade kľúčov, následkom čoho sa užívateľ nemôže dostať k svojim údajom. V tomto režime sa niektoré operácie nemôžu spracovať okamžite a sú preto naplánované ako nevybavené operácie (čakajúce na spracovanie). Takéto nevybavené operácie sa vygenerujú v prípadoch ako je pridanie alebo potlačenie skupinového prístupového kľúča v užívateľskom sklade kľúčov alebo regenerovanie súkromného kľúča. Vybaví ich vlastník skladu kľúčov.

Administračný kľúč efs_admin:

Sklad kľúčov efs_admin obsahuje špeciálny kľúč, ktorý dokáže otvoriť ľubovoľný užívateľský alebo skupinový sklad kľúčov v režime koreňového administrátora (predvolený režim).

Heslo otvárajúce tento špeciálny sklad kľúčov je uložený v skladoch kľúčov koreňového užívateľa a bezpečnostnej skupiny, keď je aktivovaný EFS. Toto heslo potom môže byť poskytnuté alebo odobraté iným užívateľom alebo skupinám pomocou príkazu **efskeymgr**. Tento kľúč, v súčinnosti s autorizáciou RBAC **aix.security.efs**, dovoľuje užívateľovi spravovať EFS (to znamená, pristupovať ku skladom kľúčov v režime koreňového administrátora).

Poznámka k efs_admin RBAC

Na systémoch, kde je povolené riadenie prístupu na základe rolí, je príkaz **efs_admin** chránený autorizáciou **aix.security.efs**.

Užívateľský sklad kľúčov:

Užívateľský sklad kľúčov je v prípade väčšiny bežných operácií spravovaný automaticky. Príkaz **efskeymgr** sa používa na úlohy údržby a na pokročilé používanie systému EFS. Užívatelia môžu pomocou príkazu **efsmgr** vytvárať šifrované súbory a adresáre. Riadenie skladu kľúčov je integrované do väčšiny príkazov user admin. Ak je do skupiny pridaný nový užívateľ, ten bude mať automaticky prístup ku skladu kľúčov skupiny.

Vlastník súboru s EFS prístupom k súboru poskytne pomocou príkazu **efsmgr** EFS prístup iným užívateľom a skupinám (je to podobné ako v prípade kontroly, ktorú majú vlastníci súborov so zoznamami ACL v systéme UNIX). Užívatelia môžu meniť svoje heslo bez dopadu na samostatné procesy, spustené pod rovnakým UID s otvoreným skladom kľúčov.

Sklad kľúčov pre Encrypted File System

Sklady kľúčov sú chránené heslom. Užívatelia si môžu zvoliť iné heslo, než aké používajú na prihlásenie sa. V takom prípade sa sklad kľúčov počas štandardného prihlásenia užívateľa neotvorí a nebude prístupný. Namiesto toho musí užívateľ sklad kľúčov ručne zaviesť pomocou príkazu **efskey** a zadať k nemu heslo.

Formát skladu kľúčov je **PKCS # 12**. Sklady kľúčov sú uchovávané v nasledujúcich súboroch:

užívateľský sklad kľúčov
/var/efs/users//keystore

skupinový sklad kľúčov
/var/efs/groups//keystore

sklad kľúčov efsadmin
/var/efs/efs_admin/keystore

Ak si užívateľ nastaví rovnaké heslo pre prihlásenie sa aj pre svoj sklad kľúčov, po prihlásení sa jeho sklad kľúčov otvorí a bude povolený.

Užívateľ môže pomocou príkazu EFS **efskeymgr** vybrať typ šifrovacieho algoritmu a dĺžku kľúča.

Prístup do skladu kľúčov dedia všetky dcérske procesy.

Podporovaná je aj správa kľúčov založená na skupinách. Ak je skupinový sklad kľúčov v chránenom režime, skupinové kľúče ku skladom kľúčov členov môžu pridávať alebo odstraňovať iba členovia skupiny. Užívateľský sklad kľúčov obsahuje súkromný kľúč užívateľa aj heslo na otvorenie skupinových skladov kľúčov užívateľa, ktoré obsahujú súkromné kľúče skupiny.

Poznámka: Sklad kľúčov EFS sa otvorí automaticky počas štandardného prihlásenie sa do systému AIX len vtedy, keď sa heslo ku skladu kľúčov užívateľa zhoduje s užívateľovým prihlasovacím heslom. Toto sa nastaví štandardne počas úvodného vytvárania skladu kľúčov užívateľa. Iné než štandardné prihlasovacie metódy do systému AIX, napríklad načítateľné autentifikačné moduly alebo zasunuteľné autentifikačné moduly, sklad kľúčov automaticky neotvoria.

Šifrovanie a dedenie

EFS je vlastnosťou J2. Voľba súborového systému **efs** musí byť nastavená na **yes** (pozri príkazy **mkfs** a **chfs**).

J2 EFS automaticky šifruje a dešifruje užívateľské údaje. Ak však užívateľ má oprávnenie na čítanie súboru chráneného EFS, ale nemá správny kľúč, potom nemôže súbor čítať bežným spôsobom; pokiaľ užívateľ nemá platný kľúč, je nemožné dešifrovať údaje.

Všetky šifrovacie funkcie vychádzajú z kernelových služieb CLiC a z užívateľských knižníc CLiC.

Štandardne nie je súborový systém J2 povolený pre EFS. Systém súborov J2 musí byť najskôr povolený pre EFS, až potom je možné aktivovať dedičnosť EFS alebo použiť EFS šifrovanie na akékoľvek užívateľské údaje. Šifrovaný súbor sa vytvorí buď explicitne príkazom **efsmgr**, alebo implicitne prostredníctvom dedenia EFS. EFS dedenie môžete aktivovať na úrovni súborového systému, na úrovni adresára, prípadne oboch.

Príkaz **ls** vypíše položky šifrovaného súboru s predchádzajúcim **e**.

Príkazy **cp** a **mv** dokážu jednoducho spracovať metaúdaje a zašifrované údaje v scenároch EFS-na-EFS a EFS-na-nie-EFS.

Príkazy **backup**, **restore** a **tar** a súvisiace príkazy dokážu zálohovať a obnoviť zašifrované údaje, vrátane metaúdajov EFS používaných na šifrovanie a dešifrovanie.

Zálohovanie a obnovenie

Je dôležité správne zvládnuť archiváciu alebo zálohovanie skladov kľúčov priradených archivovaným súborom EFS. Musíte tiež spravovať a uchovávať heslá pre sklady kľúčov, priradené archivovaným alebo zálohovaným skladom kľúčov. Ak niektorú z týchto úloh nevykonáte správne, môže dôjsť k strate údajov.

Keď zálohujete šifrované súbory EFS, môžete použiť voľbu **–Z** s príkazom **backup** na zálohovanie šifrovanej formy súboru spolu so šifrovacími metaúdajmi súboru. Súborové údaje aj metaúdaje sú chránené silným šifrovaním. Z bezpečnostného hľadiska je výhodnejšie chrániť zálohované údaje silným šifrovaním. Je nutné zálohovať sklad kľúčov vlastníka súboru a skupiny, priradené práve zálohovanému súboru. Tieto sklady kľúčov sú umiestnené v nasledujúcich súboroch:

užívateľské sklady kľúčov
/var/efs/users/*user_login*/*

skupinový sklad kľúčov
/var/efs/groups//keystore

sklad kľúčov efsadmin
/var/efs/efs_admin/keystore

Na obnovenie zálohy EFS sa používa príkaz **restore** (vykonáva sa pomocou príkazu **backup** a voľby **–Z**). Príkaz na obnovu zabezpečí, aby boli obnovené aj šifrovacie metaúdaje. Počas procesu obnovy nie je nutné obnoviť zálohované sklady kľúčov, pokiaľ užívateľ nezmenil kľúče vo svojom vlastnom sklade kľúčov. Keď užívateľ zmení svoje heslo na otvorenie skladu kľúčov, jeho interný kľúč skladu kľúčov sa nezmení. Ak chcete zmeniť interné kľúče skladu kľúčov, použite príkaz **efskeymgr**.

Ak sa užívateľov interný kľúč skladu kľúčov nezmení, užívateľ môže hneď otvoriť a dešifrovať obnovený súbor pomocou aktuálneho skladu kľúčov. Ak sa však interný kľúč zmenil, užívateľ musí otvoriť sklad kľúčov, ktorý bol zálohovaný v súvislosti so zálohovaným súborom. Takýto sklad kľúčov sa dá otvoriť príkazom **efskeymgr –o**. Príkaz **efskeymgr** vyzve užívateľa zadať heslo na otvorenie skladu kľúčov. Je to heslo, ktoré bolo použité v súvislosti so skladom kľúčov v čase zálohovania.

Predpokladajme napríklad, že sklad kľúčov užívateľa Boba bol chránený heslom **foo** (heslo ‘foo’ nie je bezpečné heslo; je tu použité len na ilustratívne účely kvôli zjednodušeniu) a Bobove zašifrované súbory boli odzálohované v januári spolu s Bobovým skladom kľúčov. V tomto príklade Bob použije **foo** aj ako prihlasovacie heslo do systému AIX. Vo februári Bob zmení svoje heslo na **bar**, následkom čoho sa aj jeho prístupové heslo ku skladu kľúčov zmení na **bar**. Ak budú v marci Bobove EFS súbory obnovené, Bob ich bude môcť otvoriť a zobraziť pomocou svojho aktuálneho skladu kľúčov a hesla, pretože nezmenil interný kľúč skladu kľúčov.

Ak však bolo nutné zmeniť interný kľúč Bobovho skladu kľúčov (pomocou príkazu **efskeymgr**), potom bude štandardne odmietnutý interný kľúč starého skladu kľúčov a ponechá sa v Bobovom sklade kľúčov. Keď užívateľ pristupuje k súboru, EFS zistí automaticky, že obnovený súbor používal starý interný kľúč a EFS potom použije na jeho dešifrovanie odmietnutý kľúč. Počas tejto istej prístupovej inštancie skonvertuje EFS súbor na používanie nového

interného kľúča. Tento proces nemá žiadny podstatný vplyv na výkon, pretože je celý spracovaný cez sklad kľúčov a šifrovacie metaúdaje súboru a nevyžaduje opätovné zašifrovanie údajov súboru.

Ak je odmietnutý interný kľúč odstránený pomocou **efskeymgr**, potom musí byť starý sklad kľúčov obsahujúci starý interný kľúč obnovený a používaný spolu so súbormi zašifrovanými s týmto interným kľúčom.

Vzniká tým otázka, ako sa dajú bezpečne uchovávať a archivovať staré heslá. Na archivovanie hesiel existujú viaceré metódy a nástroje. Vo všeobecnosti spočívajú v tom, že je potrebné mať súbor, ktorý obsahuje zoznam starých hesiel. Tento súbor je potrebné zašifrovať a chrániť ho aktuálnym skladom kľúčov, ktorý je zase chránený aktuálnymi heslami. Keďže však IT prostredia a bezpečnostné politiky sa v rôznych organizáciách môžu veľmi líšiť, je potrebné veľmi dôkladne zvážiť špecifické potreby vašej organizácie v súvislosti s bezpečnosťou a vyvinúť také bezpečnostné politiky a postupy, ktoré budú vyhovovať vášmu prostrediu.

Interný mechanizmus J2 EFS

Každý súbor chránený cez J2 EFS je asociovaný so špeciálnym rozšíreným atribútom, ktorý obsahuje metaúdaje EFS, používané na overenie platnosti šifrovacieho oprávnenia, a informácie používané na šifrovanie dešifrovanie súborov (kľúče, šifrovací algoritmus, atď.).

Obsah EA je pre J2 viditeľný. Na určenie oprávnenia na šifrovanie (riadenie prístupu) sa vyžadujú užívateľské splnomocnenia aj metaúdaje EFS pre akýkoľvek daný, systémom EFS chránený súbor.

Poznámka: Špeciálnu pozornosť treba venovať situáciám, kedy môže dôjsť k strate údajov alebo súborov (napríklad pri odstraňovaní EA súboru).

Dedenie ochrany EFS

Keď je adresár aktivovaný pre EFS, všetci jeho novo vytvorení priami potomkovia budú tiež aktivovaní pre EFS, ak to nebude ručne zmenené. Atribúty EFS rodičovského adresára majú prednosť pred atribútmi EFS súborového systému.

Rozsah dedičnosti adresára je presne jedna úroveň. Každý novo vytvorený potomok zdedí atribúty EFS svojho rodiča, ak je jeho rodič chránený systémom EFS. Existujúci potomok si zachová svoj aktuálny zašifrovaný alebo nezašifrovaný stav. Logická reťaz dedičnosti sa preruší, ak sa zmenia EFS atribúty rodiča. Tieto zmeny nie sú prenesené nižšie na existujúceho potomka adresára a musia byť na tieto adresáre použité automaticky.

Poznámky k oddielom pracovného zaťaženia

Skôr, ako môžete použiť alebo povoliť systém EFS v rámci oddielu pracovného zaťaženia, najskôr musí byť systém EFS povolený na globálnom systéme pomocou príkazu **efsenable**. Toto povolenie stačí vykonať raz. Okrem toho musia byť všetky systémy súborov, vrátane súborových systémov s aktivovaným EFS, vytvorené z globálneho systému.

Nastavenie systému EFS

Toto musíte urobiť najprv.

Všetko treba nastaviť presne takto.

1. Nainštalujte sadu súborov **clic.rte**. Sada súborov obsahuje šifrovacie knižnice a rozšírenie kernelu, ktoré vyžaduje EFS. Sada súborov **clic.rte** sa nachádza v balíku rozšírení AIX Expansion Pack.
2. Pomocou príkazu **efsenable** povoľte na systéme EFS (použite napríklad príkaz >efsenable –a). Na výzvu zadať heslo je rozumné zadať koreňové heslo. Automaticky sa vytvoria užívateľské sklady kľúčov, na čo sa užívateľ môže prihlásiť, alebo znovu prihlásiť, po spustení príkazu **efsenable**. Stačí, aby bol na systéme raz spustený príkaz **efsenable –a**, a systém je povolený pre EFS, čiže príkaz **efsenable** netreba spúšťať znova.
3. Vytvorte súborový systém povolený pre EFS použitím voľby **–a efs=yes**. Napríklad: crfs -v jfs2 -m /foo –A yes -a efs=yes -g rootvg -a size=20000
4. Po pripojení súborového systému zapnite na súborovom systéme povolenom pre EFS dedenie šifrovania. Použite na to príkaz **efsmgr**. Podľa predchádzajúceho príkladu, kde bol vytvorený súborový systém **/foo**, spustite príkaz: efsmgr –s –E /foo. Umožňuje, aby každý súbor, vytvorený a používaný v tomto súborovom systéme, mohol byť šifrovaný súbor.

Od tohto momentu vždy, keď užívateľ alebo proces s otvoreným skladom kľúčov vytvorí v tomto súborovom systéme súbor, tento súbor bude šifrovaný. Keď užívateľ alebo súbor číta takýto súbor, automaticky sa dešifruje pre tých užívateľov, ktorí majú oprávnenie na prístup k nemu.

Viac informácií nájdete v častiach venovaných týmto príkazom:

- **chfs**, **chgroup**, **chuser**, **cp**, **efsenable**, **efskeymgr**, **efsmgr**, **lsuser**, **ls**, **mkgroup**, **mkuser** a **mv**
- a k súborom /etc/security/group a /etc/security/user

Vzdialený prístup k zásobníkom kľúčov EFS (Encrypted File System)

V podnikovom prostredí môžete centralizovať zásobníky kľúčov EFS (Encrypted File System). Keď uložíte databázy, ktoré riadia zásobníky kľúčov v každom systéme nezávisle, zásobníky kľúčov môže byť náročné riadiť. Centralizovaný zásobník kľúčov EFS v systéme AIX umožňuje uložiť databázy zásobníkov kľúčov užívateľov a skupín v LDAP (Lightweight Directory Access Protocol), aby ste zásobník kľúčov EFS mohli riadiť centrálne.

Súvisiace koncepty:

“Lightweight Directory Access Protocol” na strane 146
Lightweight Directory Access Protocol (LDAP) definuje štandardnú metódu prístupu a aktualizácie informácií v adresári (databáze) lokálne alebo vzdialene v modeli servera klienta.

Prehľad vzdialeného prístupu k zásobníkom kľúčov EFS (Encrypted File System):

Získajte informácie o databázach EFS (Encrypted File System), schopnosti LDAP pre príkazy EFS a jedinečnom prístupe k zásobníku kľúčov.

Na server LDAP môžete uložiť všetky databázy skladov kľúčov EFS systému AIX vrátane nasledujúcich databáz EFS:

- Zásobník kľúčov užívateľa
- Zásobník kľúčov skupiny
- Zásobník kľúčov administrátora
- Cookie

Operačný systém AIX poskytuje pomocné programy, ktoré vám pomôžu pri vykonávaní nasledujúcich úloh správy:

- Exportovať lokálne údaje zásobníka kľúčov na LDAP server
- Konfigurovať klienta na používanie údajov zásobníka kľúčov EFS v LDAP
- Riadiť prístup k údajom zásobníka kľúčov EFS
- Riadiť údaje LDAP z klientskeho systému

Všetky príkazy riadenia databázy zásobníka kľúčov EFS sú povolené na používanie databázy zásobníka kľúčov LDAP. Ak v súbore /etc/nscontrol.conf nie je zadané poradie vyhľadávania v celom systéme, operácie zásobníka kľúčov budú závisle od atribútu efs_keystore_access užívateľa a skupiny. Ak ste atribút efs_keystore_access nastavili na ldap, príkazy EFS vykonajú operácie zásobníka kľúčov v zásobníku kľúčov LDAP.

Nasledujúca tabuľka popisuje zmeny v príkazoch EFS pre LDAP.

Tabuľka 12. Schopnosť príkazu EFS pre LDAP

Command	Informácie LDAP
Ľubovoľný príkaz EFS	Keď nastavíte atribút efs_keystore_access na ldap, na vykonanie operácií zásobníka kľúčov v LDAP nemusíte použiť špeciálnu voľbu -L *domain* s ktorýmkoľvek príkazom.
efskeymgr	Obsahuje voľbu -L load_module, aby ste mohli vykonať explicitné operácie zásobníka kľúčov v LDAP.
efsenable	Obsahuje voľbu -d Basedn, aby ste mohli vykonať úvodné nastavenie v LDAP na umiestnenie zásobníka kľúčov EFS. Úvodné nastavenie zahŕňa pridanie základných jedinečných názvov (DN) pre zásobník kľúčov EFS a vytvorenie štruktúry lokálnych adresárov (/var/efs/).

Tabuľka 12. Schopnosť príkazu EFS pre LDAP (pokračovanie)

Command	Informácie LDAP
efskstoldif	Vygeneruje údaje zásobníka kľúčov EFS pre LDAP z nasledujúcich databáz v lokálnom systéme: • /var/efs/users/*username*/keystore • /var/efs/groups/*groupname*/keystore • /var/efs/efs_admin/keystore • Cookie, ak existujú, pre všetky zásobníky kľúčov

Všetky položky zásobníka kľúčov musia byť jedinečné. Každá položka zásobníka kľúčov priamo zodpovedá jedinečnému názvu položky, ktorá obsahuje meno užívateľa a názov skupiny. Systém dotazuje ID užívateľov (uidNumber), ID skupín (gidNumber) a jedinečné názvy. Dotaz je úspešný, keď sa mená užívateľov a názvy skupín zhodujú s jedinečnými názvami. Skôr ako vytvoríte alebo migrujete položky zásobníka kľúčov EFS na LDAP musíte skontrolovať, či sú ID a mená užívateľov a názvy skupín v systéme jedinečné.

Súvisiace úlohy:

"Export údajov zásobníka kľúčov EFS (Encrypted File System) do LDAP"
LDAP server musíte zaplniť údajmi zásobníka kľúčov, aby ste mohli LDAP používať ako centralizovaný archív pre zásobník kľúčov EFS (Encrypted File System).

"Konfigurácia klienta LDAP pre zásobník kľúčov EFS (Encrypted File System)"
Ak chcete použiť údaje zásobníka kľúčov EFS (Encrypted File System), ktoré sú uložené v LDAP, systém musíte konfigurovať ako klienta LDAP.

Export údajov zásobníka kľúčov EFS (Encrypted File System) do LDAP:

LDAP server musíte zaplniť údajmi zásobníka kľúčov, aby ste mohli LDAP používať ako centralizovaný archív pre zásobník kľúčov EFS (Encrypted File System).

Skôr ako vytvoríte alebo migrujete položky zásobníka kľúčov EFS na LDAP musíte skontrolovať, či sú ID a mená užívateľov a názvy skupín v systéme jedinečné.

Ak chcete zaplniť LDAP server údajmi zásobníka kľúčov EFS, postupujte podľa týchto krokov:

1. Schému zásobníka kľúčov EFS pre LDAP nainštalujte na LDAP server:
   a. Získajte schému skladu kľúčov EFS pre LDAP zo súboru /etc/security/ldap/sec.ldif v systéme AIX.
   b. Príkazom **ldapmodify** aktualizujte schému LDAP servera so schémou zásobníka kľúčov EFS pre LDAP.
2. Pomocou príkazu **efskstoldif** prečítajte údaje v lokálnych súboroch zásobníka kľúčov EFS a pripravte výstup údajov vo formáte, ktorý bude vyhovovať LDAP. Ak chcete zachovať jedinečný prístup k zásobníku kľúčov, porozmýšľajte, či údaje zásobníka kľúčov EFS, ktoré sa nachádzajú v LDAP pod rovnakým jedinečným rodičovským názvom (DN), neumiestnite ako údaje užívateľov a skupín.
3. Údaje uložte do súboru.
4. Ak chcete zaplniť LDAP server údajmi zásobníka kľúčov, spustite príkaz **ldapadd -b**.

Súvisiace koncepty:

"Prehľad vzdialeného prístupu k zásobníkom kľúčov EFS (Encrypted File System)" na strane 169
Získajte informácie o databázach EFS (Encrypted File System), schopnosti LDAP pre príkazy EFS a jedinečnom prístupe k zásobníku kľúčov.

Konfigurácia klienta LDAP pre zásobník kľúčov EFS (Encrypted File System):

Ak chcete použiť údaje zásobníka kľúčov EFS (Encrypted File System), ktoré sú uložené v LDAP, systém musíte konfigurovať ako klienta LDAP.

Ak chcete konfigurovať klienta LDAP pre zásobník kľúčov EFS, postupujte podľa nasledujúcich krokov:

1. Pomocou príkazu **/usr/sbin/mksecldap** nakonfigurujte systém ako klienta LDAP. Príkaz **mksecldap** dynamicky vyhľadá označený LDAP server, aby zistil umiestnenie údajov zásobníka kľúčov EFS. Výsledky potom uloží do súboru /etc/security/ldap/ldap.cfg. Príkaz **mksecldap** stanoví umiestnenie pre údaje zásobníka kľúčov užívateľa, skupiny, administrátora a efscookies.
2. Vykonajte jeden z nasledujúcich krokov, ktorým aktivujete LDAP ako doménu vyhľadávania pre údaje zásobníka kľúčov EFS:
   - Atribút **efs_keystore_access** užívateľa a skupiny nastavte na **file** alebo **ldap**.
   - Pomocou súboru /etc/nscontrol.conf stanovte poradie vyhľadávania pre zásobník kľúčov na systémovej úrovni. Nasledujúca tabuľka uvádza príklad.

Tabuľka 13. Vzorová konfigurácia pre súbor /etc/nscontrol.conf

Atribút	Description	Poradie vyhľadávania (secorder)
efsusrkeystore	Toto poradie vyhľadávania je bežné pre všetkých užívateľov.	LDAP, súbory
efsgrpkeystore	Toto poradie vyhľadávania je bežné pre všetky skupiny.	súbory, LDAP
efsadmkeystore	Toto poradie vyhľadávania vyhľadá zásobník kľúčov administrátora pre ľubovoľný cieľový zásobník kľúčov.	LDAP, súbory

Upozornenie: Konfigurácia definovaná v súbore /etc/nscontrol.conf nahradí všetky hodnoty nastavené pre atribút **efs_keystore_access** užívateľa a skupiny. To isté platí pre atribút **efs_adminks_access** užívateľa.

Démon klienta /usr/sbin/secldapclntd po konfigurácii systému ako klienta LDAP a aktivácii LDAP ako domény vyhľadávania pre údaje zásobníka kľúčov EFS získa údaje zásobníka kľúčov z LDAP servera vždy keď vykonáte operácie zásobníka kľúčov LDAP.

Súvisiace koncepty:

“Prehľad vzdialeného prístupu k zásobníkom kľúčov EFS (Encrypted File System)” na strane 169
Získajte informácie o databázach EFS (Encrypted File System), schopnosti LDAP pre príkazy EFS a jedinečnom prístupe k zásobníku kľúčov.

Public Key Cryptography Standards #11

Podsystém PKCS #11 (Public Key Cryptography Standards #11) poskytuje aplikáciám metódu prístupu na hardvérové zariadenia (tokeny) bez ohľadu na typ zariadenia.

Obsah tejto časti vyhovuje verzii 2.20 štandardu PKCS #11.

Podsystém PKCS #11 využíva nasledujúce komponenty:

- Zdieľaný objekt API (/usr/lib/pkcs11/ibm_pks11.so) je poskytovaný ako všeobecné rozhranie k ovládaču zariadenia, ktorý podporuje štandard PKCS #11. Tento vrstvený dizajn povoľuje nové zariadenia PKCS #11, keď sú k dispozícii, bez rekompilácie existujúcich aplikácií.
- Ovládač zariadenia PKCS #11, ktorý poskytuje aplikáciám podobné schopnosti ako tie, ktoré sú poskytované iným komponentom jadra, napríklad EFS (Encrypted File System) alebo IPSec (IP Security).
- Ak platforma podporuje zariadenie kryptografického koprocesora, ovládač zariadenia PKCS #11 využíva hardvérovú akceleráciu, ktorá je k dispozícii s operáciami AES (Advanced Encryption Standard), SHA (Secure Hash Algorithm) a HMAC (Hash Message Authentication Code). Výkonnosť môžete zlepšiť povolením afinity sieťovej pamäte.

Súvisiace informácie:

Podpora afinity pamäte AIX

IBM 4758 Model 2 Cryptographic Coprocessor

IBM 4758 Model 2 Cryptographic Coprocessor poskytuje bezpečné výpočtové prostredie.

Skôr než sa pokúsite o konfiguráciu podsystému PKCS #11 skontrolujte, či bol adaptér správne nakonfigurovaný podporovaným mikrokódom.

IBM 4960 Cryptographic Accelerator

IBM 4960 Cryptographic Accelerator poskytuje prostriedky na offloading kryptografických transakcií. Pred vykonaním pokusu o konfiguráciu podsystému PKCS #11 skontrolujte správnosť konfigurácie adaptéra.

Overovanie IBM 4758 Model 2 Cryptographic Coprocessor pre použitie s podsystémom Public Key Cryptography Standards #11:

Subsystém PKCS #11 je určený na automatické rozpoznávanie adaptérov, ktoré podporujú volania PKCS #11 počas inštalácie alebo opätovného zavedenia. Z tohto dôvodu nebude žiadny nesprávne nakonfigurovaný adaptér IBM 4758 Model 2 Cryptographic Coprocessor dostupný z rozhrania PKCS #11 a volania, odoslané do tohto adaptéra, zlyhajú.

Ak si chcete overiť, či je váš adaptér správne nastavený, postupujte nasledovne:

1. Ak sa chcete uistiť, že softvér pre adaptér je správne nainštalovaný, zadajte nasledujúci príkaz:
   ```
   lsdev -Cc adapter | grep crypt
   ```

   Ak sa IBM 4758 Model 2 Cryptographic Coprocessor nebude nachádzať na konečnom zozname, skontrolujte, či je karta správne osadená a či je ovládač podporného zariadenia správne nainštalovaný.
2. Pomocou nasledovného príkazu zistite, či bol do karty zavedený príslušný firmvér:
   ```
   csufclu /tmp/l ST device_number_minor
   ```

   Overte, či je pre obrázok segmentu 3 načítaná aplikácia PKCS #11. Ak nie je načítaná, v dokumentácii k adaptéru vyhľadajte najnovší mikrokód a podľa inštrukcií ho nainštalujte.

   Poznámka: Ak tento pomocný program nie je k dispozícii, potom nebol nainštalovaný podporný softvér.

Overenie IBM 4960 Model 2 Cryptographic Accelerator na použitie s podsystémom Public Key Cryptography Standards #11:

Subsystém PKCS #11 je určený na automatické rozpoznávanie adaptérov, ktoré podporujú volania PKCS #11 počas inštalácie alebo opätovného zavedenia. Z toho dôvodu nebude IBM 4960 Cryptographic Accelerator, ktorý nie je riadne nakonfigurovaný, dostupný z rozhrania PKCS #11 a volania zaslané do adaptéra budú neúspešné.

Ak sa chcete uistiť, že softvér pre adaptér je správne nainštalovaný, zadajte nasledujúci príkaz:
```
lsdev -Cc adapter | grep ica
```

Ak nie je IBM 4960 Cryptographic Accelerator zahrnutý do výsledného zoznamu, skontrolujte správne umiestnenie karty a správnu inštaláciu podporného ovládača zariadenia.

Konfigurácia podsystému Public Key Cryptography Standards #11

Subsystém PKCS #11 automaticky rozpozná zariadenia podporujúce štandard PKCS #11. Aby však mohli tieto zariadenia používať niektoré aplikácie, vyžaduje sa prvotné nastavenie.

Tieto úlohy možno vykonať pomocou rozhrania API (vytvorením aplikácie PKCS #11) alebo pomocou rozhrania SMIT. Voľby rozhrania SMIT pre štandard PKCS #11 sú dostupné kliknutím na položku **Manage the PKCS11 subsystem** v hlavnej ponuke SMIT alebo pomocou rýchlej cesty **smit pkcs11**.

Inicializácia tokenu:

Každý adaptér alebo token PKCS #11 musí byť najprv inicializovaný, a až potom môže byť úspešne použitý.

Táto procedúra inicializácie zahŕňa nastavenie jedinečného návestia pre token. Toto návestie umožňuje aplikáciám jednoznačne identifikovať token. Návestia by sa preto nemali opakovať. Rozhranie API však neoveruje, či sú návestia jedinečné. Inicializáciu je možné uskutočniť pomocou aplikácie PKCS #11 alebo s oprávnením systémového administrátora pomocou rozhrania SMIT. Ak pre daný token existuje PIN bezpečnostného pracovníka (SO PIN), predvolená hodnota je nastavená na 87654321. Aby sa zaručila bezpečnosť podsystému PKCS #11, túto hodnotu by ste mali po inicializácii zmeniť.

Inicializácia tokenu:

1. Zadaním príkazu smit pkcs11 spustite obrazovku pre správu tokenov.
2. Vyberte voľbu **Initialize a Token** .
3. Zo zoznamu podporovaných adaptérov vyberte adaptér PKCS #11.
4. Svoj výber potvrďte stlačením klávesu Enter.

   Poznámka: Týmto sa vymažú všetky informácie v tokene.
5. Zadajte SO PIN a jedinečné návestie tokenu.

Ak je zadaný správny PIN, po vykonaní príkazu sa uskutoční inicializácia alebo opätovná inicializácia adaptéra.

Nastavenie PIN kódu bezpečnostného technika:

Ak chcete zmeniť štandardnú hodnotu PIN bezpečnostného technika, postupujte nasledovne.

Ak chcete zmeniť štandardnú hodnotu PIN:

1. Zadajte príkaz smit pkcs11.
2. Vyberte voľbu **Set the Security Officer PIN**.
3. Vyberte inicializovaný adaptér, pre ktorý chcete PIN nastaviť.
4. Zadajte aktuálny PIN a nový PIN.
5. Overte novú hodnotu PIN kódu.

Inicializácia užívateľského PIN kódu:

Po inicializácii tokenu, bude možno nutné nastaviť PIN kód užívateľa, aby umožnil aplikáciám prístup na objekty tokenu.

Pozrite si dokumentáciu pre vaše konkrétne zariadenie, aby ste zistili, či zariadenie vyžaduje, aby sa užívateľ prihlásil pred pristúpením na objekty.

Incializácia PIN kódu užívateľa:

1. Zadaním príkazu smit pkcs11 spustite obrazovku pre správu tokenov.
2. Vyberte voľbu **Initialize the User PIN**.
3. Zo zoznamu podporovaných adaptérov vyberte adaptér PKCS #11.
4. Zadajte SO PIN a PIN užívateľa.
5. Overte hodnotu PIN kódu užívateľa.
6. PIN užívateľa je nutné zmeniť počas overovania.

Resetovanie PIN užívateľa:

Ak chcete resetovať PIN užívateľa, môžete PIN opakovane inicializovať pomocou SO PIN, alebo môžete PIN užívateľa nastaviť použitím existujúceho PIN užívateľa.

Ak chcete PIN resetovať:

1. Zadaním príkazu smit pkcs11 spustite obrazovku pre správu tokenov.

2. Vyberte voľbu **Set the User PIN**.
3. Vyberte inicializovaný adaptér, pre ktorý chcete PIN užívateľa nastaviť.
4. Zadajte aktuálnu aj novú hodnotu PIN kódu užívateľa.
5. Overte novú hodnotu PIN kódu užívateľa.

Použitie Public Key Cryptography Standards #11

Aby mohla aplikácia podsystém PKCS #11 používať, démon manažéra slotov podsystému musí byť spustený a aplikácia sa musí zaviesť do objektu zdieľaného API rozhraniami.

Manažér slotov sa bežne spúšťa v čase zavedenia pomocou **inittab**, ktorý volá skript /etc/rc.pkcs11. Tento skript overí adaptéry v systéme pred spustením procesu typu démon pre správcu Slot manager. Výsledkom spustenia tohto skriptu je, že démon pre správcu Slot manager nie je dostupný pred prihlásením užívateľa do systému. Po spustení procesu typu démon systém aplikuje všetky vykonané zmeny počtu a typov podporovaných adaptérov bez zásahu systémových administrátorov.

Zdieľaný objekt rozhrania API možno načítať buď prepojením objektu počas relácie spustenia alebo pomocou stratégie odloženého rozlíšenia symbolov. Aplikácia môže získať zoznam funkcií PKCS #11 napríklad nasledovným spôsobom:

```
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);
```

Nástroje Public Key Cryptography Standards #11

Na správu šifrovacích systémov v rámci operačného systému AIX sú k dispozícii dva nástroje: nástroj PKCS #11 Key Management a nástroj PKCS #11 Administration. K týmto nástrojom môžete získať prístup použitím buď grafického užívateľského rozhrania, založeného na knižnici Curses alebo rozhrania príkazového riadka.

Poznámka: Funkcie na zjednodušenie ovládania v nástrojoch šifrovacieho rámca AIX vyžadujú použitie funkcií dávkového spracovania. Podrobné informácie o používaní funkcií dávkového spracovania pre zjednodušenie ovládania nájdete v časti “Dávkové spracovanie” na strane 176.

Nástroj PKCS #11 Key Management predstavuje centralizovaný nástoj na správu kľúčov, certifikátov a údajov PKCS #11 v operačnom systéme AIX. Objekty spravované týmto nástrojom sú uložené v podporovaných poskytovateľoch PKCS #11, ako napríklad rodina šifrovacích adaptérov IBM (ako sú IBM 4758, 4960 a 4764), alebo rámec AIX Cryptographic Framework. Použitím nástroja PKCS #11 Key Management môžete vykonávať rôzne operácie. Medzi tieto operácie patrí vytvorenie požiadavky na podpísanie certifikátu PKCS #10 (CSR) alebo vygenerovanie samopodpísaných certifikátov. Okrem toho môžete tento nástroj používať na vyhľadávanie, zobrazovanie, vymazávanie, import, export a zálohovanie údajov objektov PKCS #11, ako aj na prenos údajov objektov PKCS #11 medzi symbolmi PKCS #11. Verziu grafického užívateľského rozhrania tohto nástroja môžete spustiť použitím príkazu **p11km**. Tento nástroj zavedie všetky dostupné symboly PKCS #11. Podrobnosti o týchto symboloch si môžete pozrieť použitím kurzorových klávesov, pomocou ktorých sa môžete posúvať po zozname symbolov smerom nahor aj nadol. Ak chcete vybrať niektorý symbol, použitím kurzorových klávesov ho zvýraznite a stlačte kláves Enter. Verziu príkazového riadka tohto nástroja môžete spustiť použitím tohto príkazu:

```
p11km -b <batchfile>
```

Nástroj PKCS #11 Administration predstavuje centralizovaný nástroj na správu rámca AIX PKCS #11 Cryptographic Framework. Tento nástroj umožňuje administrátorom a správcom bezpečnosti spravovať tokeny riadené rámcom AIX Cryptographic Framework. Pomocou tohto nástroja môžete inicializovať, vytvárať a ničiť tokeny PKCS #11, spravovať sloty, vynulovať heslá užívateľov, potvrdzovať odstránenia objektov, určiť dôveryhodnosť objektov a vykonávať ladenie rámca AIX Cryptographic Framework v oblasti výkonnosti aj všeobecnej správy. Verziu grafického užívateľského rozhrania tohto nástroja môžete spustiť použitím príkazu **p11admin**. Tento nástroj zavedie všetky dostupné symboly PKCS #11. Podrobnosti o týchto symboloch si môžete pozrieť použitím kurzorových klávesov, pomocou ktorých sa môžete posúvať po zozname symbolov smerom nahor aj nadol. Ak chcete vybrať niektorý symbol, použitím kurzorových klávesov ho zvýraznite a stlačte kláves Enter. Verziu príkazového riadka tohto nástroja môžete spustiť použitím tohto príkazu:

```
p11admin -b <batchfile>
```

Profily príkazu:

Nástroje AIX Cryptographic Framework používajú knižnicu OpenSSL na analýzu konfiguračných súborov, ktoré sa používajú pri vytváraní vlastných profilov. Pomocou týchto profilov môžete nastaviť atribúty nástrojov, napríklad farby grafického užívateľského rozhrania pre príkazy **p11km** a **p11admin**.

Použitím formátu súboru, špecifikovaného v “Dávkové spracovanie” na strane 176, môžete vytvoriť a upraviť nasledujúce súbory profilov a pomocou nich prispôsobiť grafické užívateľské rozhranie.

Poznámka: Súbory profilov po ich vytvorení pomenujte a uložte ich do svojho domovského adresára takto:

```
$HOME/.p11km
```

```
$HOME/.p11admin
```

Podporované sú nasledujúce atribúty farieb grafického užívateľského rozhrania:

```
action_name = "GUI_COLORS"
gui_fg_color = "<color name>" ## Foreground Color
gui_bg_color = "<color name>" ## Background Color
gui_vc_color = "<color name>" ## View Content Color
```

Kde <*color name*> je niektorá z týchto hodnôt:

- LIGHT GRAY
- WHITE
- BLACK
- DARK GRAY
- RED
- LIGHT RED
- YELLOW
- ORANGE or BROWN
- GREEN
- LIGHT GREEN
- BLUE
- LIGHT BLUE
- CYAN
- LIGHT CYAN
- MAGENTA
- LIGHT MAGENTA

Príklad: p11km profile ($HOME/.p11km)

```
[p11km_cmd]
gui_fg_color = "RED"
gui_bg_color = "BLACK"
gui_vc_color = "WHITE"
```

Example: p11admin Profile ($HOME/.p11admin)

```
[p11admin_cmd]
gui_fg_color = "BLUE"
gui_bg_color = "LIGHT GRAY"
gui_vc_color = "BLACK"
```

Dávkové spracovanie:

Spustením príkazov na dávkové spracovanie z príkazového riadka môžete vykonávať rovnaké úlohy, aké sú k dispozícii vo verziách grafického užívateľského rozhrania nástrojov PKCS #11.

Formát príkazu pre nástroj PKCS #11 Key Management (p11km) je takýto:

```
p11km -b <batchfile>
```

Formát príkazu pre nástroj PKCS #11 Key Administration (p11admin) je takýto:

```
p11admin -b <batchfile>
```

Keďže tieto nástroje používajú na analýzu dávkových súborov knižnicu OpenSSL, formát dávkových súborov sa riadi typickým formátom konfiguračného súboru OpenSSL. Každá sekcia je osobitný príkaz a páry hodnôt atribútov poskytujú informácie, vyžadované pre spracovanie. Príkaz každej sekcie je dávkovo spracovaný v poradí zhora nadol. Ak individuálny dávkový príkaz zlyhá, je vytlačená chyba a dávkové spracovanie sa ukončí bez spracovania príkazov nasledujúcich sekcií.

Nasleduje príklad formátu konfiguračného súboru OpenSSL.

```
[section1]
attribute1 = "value1"
attribute2 = "value2"
...
attributeN = "valueN"
[section2]
attribute1 = "value1"
attribute2 = "value2"
...
attributeN = "valueN"
...
...
[sectionN]
attribute1 = "value1"
attribute2 = "value2"
...
attributeN = "valueN"
```

Ak chcete zabezpečiť, aby sekcie príkazov nástroja PKCS #11 koexistovali so sekciami konfiguračného súboru OpenSSL, pre sekcie PKCS #11 použite tieto predpony:

Nástroj p11km
p11km_cmd

Nástroj p11admin
p11admin_cmd

Každá sekcia p11km_cmd alebo p11admin_cmd musí obsahovať len jeden atribút action_name s reťazcovou hodnotou, ktorá identifikuje konkrétny príkaz, priradený k sekcii. Najjednoduchším príkladom je súbor, ktorý obsahuje jednu sekciu príkazu, popisujúcu príkaz, ktorý nemá ďalšie parametre. Nasleduje príklad postupu pri použití nástroja p11km na spustenie dávkového príkazu, ktorý vypíše zoznam dostupných symbolov PKCS #11 v systéme:

```
[p11km_cmd_list_my_tokens]
action_name="LIST_TOKENS"
```

Každý dávkový príkaz podporuje voliteľný booleovský atribút:

```
start_gui="<boolean>"
```

Ak spustíte dávkový príkaz, ktorý obsahuje booleovský atribút s hodnotou TRUE, po dokončení tohto príkazu sa ukončí dávkové spracovanie a spustí sa grafické užívateľské rozhranie.

Poznámka: Ak dávkový súbor obsahuje príkaz, ktorého súčasťou je voliteľný atribút **start_gui**, žiadne dávkové príkazy, ktoré sú vypísané po ňom, nebudú spracované.

Dávkové príkazy:

Dávkové príkazy umožňujú prístup k nástrojom PKCS #11 z príkazového riadka.

V nástroji PKCS #11 Key Management sú k dispozícii nasledujúce dávkové príkazy (p11km).

Poznámka: Ak chcete použiť dávkové príkazy, postupujte takto:

1. Podľa postupu, popísaného v “Dávkové spracovanie” na strane 176, vytvorte a upravte dávkový súbor.
2. Vytvorte nové sekcie p11km_cmd, obsahujúce atribúty pre dávkové príkazy, ktoré chcete použiť.

Vypísať zoznam dostupných symbolov PKCS #11

Vygeneruje správu a zobrazí informácie o symboloch a slotoch pre dostupné symboly PKCS #11.

Vyžadované atribúty

```
action_name = "LIST_TOKENS"
```

Voliteľné atribúty

```
start_gui = "<boolean>"
```

Kde <*boolean*> je buď TRUE alebo FALSE

Príklad

```
[p11km_cmd_list_tokens]
action_name = "LIST_TOKENS"
```

Vypísať zoznam dostupných mechanizmov PKCS#11

Vygeneruje správu a zobrazí dostupné mechanizmy PKCS #11, ktoré sú podporované konkrétnym symbolom PKCS #11 (identifikované na základe hodnôt atribútov ovládača a slotu).

Vyžadované atribúty

```
action_name = "LIST_MECHANISMS"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
```

Kde <*číslo slotu*> je hodnota kladného celého čísla a <*názov ovládača*> je niektorá z nasledujúcich hodnôt:

Hodnota	Description
AIX	Rámec Cryptographic Framework operačného systému AIX
IBM_4758_4960	Adaptéry šifrovacieho hardvéru IBM 4758/4960
IBM_4764	Adaptéry šifrovacieho hardvéru IBM 4764
Iné	Ak zadáte OTHER, musíte zadať aj atribút **p11_driver_path**.

Voliteľné atribúty

```
start_gui = "<boolean>"
```

Doplnkové atribúty

```
p11_driver_path = "<cesta k ovládaču PKCS#11>"
```

Pričom <*cesta k ovládaču PKCS#11*> predstavuje úplnú cestu v systéme UNIX k súboru knižnice PKCS #11, ktorá sa použije pri vykonaní príkazu. Tento atribút je možné zadať len v prípade, ak je atribút **p11_driver** nastavený na OTHER.

Príklad

```
[p11km_cmd_list_4764_slot_0_mechs]
action_name = "LIST_MECHANISMS"
p11_driver = "IBM_4764"
p11_slot = "0"
start_gui = "TRUE"
```

Vypísať zoznam dostupných objektov PKCS #11

Vygeneruje správu a zobrazí dostupné objekty PKCS #11, ktoré sú podporované symbolom PKCS #11 (identifikované na základe hodnôt atribútov ovládača a slotu).

Vyžadované atribúty

```
action_name = "LIST_OBJECTS"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
```

Voliteľné atribúty

```
p11_login = "<boolean>"
p11_label = "<reťazec>"
p11_class = "<PKCS#11 Object Class>"
p11_private = "<boolean>"
p11_trusted = "<boolean>"
p11_sensitive = "<boolean>"
start_gui = "<boolean>"
```

Kde <*PKCS#11 Object Class*> je niektorá z nasledujúcich hodnôt, ako je zadefinované v špecifikácii PKCS #11 z RSA:

```
CKO_DATA
CKO_CERTIFICATE
CKO_PUBLIC_KEY
CKO_PRIVATE_KEY
CKO_SECRET_KEY
CKO_HW_FEATURE
CKO_DOMAIN_PARAMETERS
CKO_MECHANISM
CKO_VENDOR_DEFINED
```

Príklad

```
[p11km_cmd_list_private_objs]
action_name = "LIST_OBJECTS"
p11_login = "TRUE"
p11_private = "TRUE"
p11_driver = "AIX"
p11_slot = "5"
```

Zmeniť PIN užívateľa symbolu PKCS #11:

Zmení PIN užívateľa symbolu PKCS #11, ktorý sa používa pri prihlasovaní do symbolu.

Vyžadované atribúty

```
action_name = "CHANGE_USER_PIN"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
```

Voliteľné atribúty

```
start_gui = "<boolean>"
```

Príklad

```
[p11km_cmd_change_my_pin]
action_name = "CHANGE_USER_PIN"
p11_slot = "1337"
p11_driver = "IBM_4764"
```

Vymazať objekty PKCS #11

Vymaže objekty PKCS #11. Objekty budú vymazané na základe číslovaného zoznamu objektov, ktorý je výsledkom spustenia príkazu **LIST_OBJECTS** a použitia rovnakej šablóny s týmito atribútmi:

```
p11_label = "<reťazec>"
p11_class = "<PKCS#11 Object Class>"
p11_private = "<boolean>"
p11_trusted = "<boolean>"
p11_sensitive = "<boolean>"
p11_login = "<boolean>"
```

Upozornenie: Keďže stav a konzistentnosť symbolu sa medzi dávkovými procesmi nezachováva, objekty môžu byť neúmyselne vymazané. Vypísané poradie objektov sa zmení, ak sú objekty pridané alebo vymazané inými procesmi, ktoré bežia na rovnakom symbole v čase medzi pôvodným vypísaním objektu a jeho vymazaním.

Vyžadované atribúty

```
action_name = "DELETE_OBJECTS"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
p11_objects = "<CSV>"
```

Kde <*CSV*> je buď slovo ALL (všetky objekty symbolov) alebo zoznam hodnôt kladných celých čísel, oddelených čiarkami, ktorý zodpovedá objektom v očíslovanom poradí výskytu použitím nasledujúcich voliteľných atribútov.

Voliteľné atribúty

```
p11_label = "<reťazec>"
p11_class = "<PKCS#11 Object Class>"
p11_private = "<boolean>"
p11_trusted = "<boolean>"
p11_sensitive = "<boolean>"
p11_login = "<boolean>"
start_gui = "<boolean>"
```

Príklad

```
[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"
```

Presunúť objekty PKCS #11:

Presunie objekty PKCS #11. Objekty budú presunuté na základe číslovaného zoznamu objektov, ktorý je výsledkom spustenia príkazu **LIST_OBJECTS** a použitia rovnakej šablóny.

Upozornenie: Keďže stav a konzistentnosť symbolu sa medzi dávkovými procesmi nezachováva, objekty môžu byť neúmyselne presunuté. Vypísané poradie objektov sa zmení, ak sú objekty pridané alebo vymazané inými procesmi, ktoré bežia na rovnakom symbole v čase medzi pôvodným vypísaním objektu a jeho presunutím.

Vyžadované atribúty

```
action_name = "MOVE_OBJECTS"
########################################
###### Source Token Identification: ######
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
########################################
```

```
###### Target Token Identification: ######
p11_driver_target = "<názov ovládača>"
p11_slot_target = "<číslo slotu>"
########################################
###### Objects being moved to target: ######
p11_objects = "<CSV>"
```

Voliteľné atribúty

```
p11_label = "<reťazec>"
p11_class = "<PKCS#11 Object Class>"
p11_private = "<boolean>"
p11_trusted = "<boolean>"
p11_sensitive = "<boolean>"
p11_login = "<boolean>"
start_gui = "<boolean>"
```

Príklad

```
[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"
```

Skopírovať objekty PKCS #11

Skopíruje objekty PKCS #11. Objekty budú skopírované na základe číslovaného zoznamu objektov, ktorý je výsledkom spustenia príkazu **LIST_OBJECTS** a použitia rovnakej šablóny.

Upozornenie: Keďže stav a konzistentnosť symbolu sa medzi dávkovými procesmi nezachováva, objekty môžu byť neúmyselne skopírované. Vypísané poradie objektov sa zmení, ak sú objekty pridané alebo vymazané inými procesmi, ktoré bežia na rovnakom symbole v čase medzi pôvodným vypísaním objektu a jeho skopírovaním.

Vyžadované atribúty

```
action_name = "COPY_OBJECTS"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
p11_driver_target = "<názov ovládača>"
p11_slot_target = "<číslo slotu>"
p11_objects = "<CSV>"
```

Voliteľné atribúty

```
p11_label = "<reťazec>"
p11_class = "<PKCS#11 Object Class>"
p11_private = "<boolean>"
p11_trusted = "<boolean>"
p11_sensitive = "<boolean>"
p11_login = "<boolean>"
start_gui = "<boolean>"
```

Príklad

```
[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" ## REQUIRED FOR PRIVATE OBJECT MGT.
```

Exportovať a zálohovať objekty PKCS #11 do súboru

Vyexportuje a zálohuje objekty PKCS #11. Objekty budú vyexportované a zálohované na základe číslovaného zoznamu objektov, ktorý je výsledkom spustenia príkazu **LIST_OBJECTS** a použitia rovnakej šablóny.

Upozornenie: Keďže stav a konzistentnosť symbolu sa medzi dávkovými procesmi nezachováva, objekty môžu byť neúmyselne vyexportované. Vypísané poradie objektov sa zmení, ak sú objekty pridané alebo vymazané inými procesmi, ktoré bežia na rovnakom symbole v čase medzi pôvodným vypísaním objektu a jeho vyexportovaním.

Vyžadované atribúty

```
action_name = "EXPORT_OBJECTS"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
p11_object_file = "<názov súboru>"
p11_objects = "<CSV>"
```

Voliteľné atribúty

```
p11_label = "<reťazec>"
p11_class = "<PKCS#11 Object Class>"
p11_private = "<boolean>"
p11_trusted = "<boolean>"
p11_sensitive = "<boolean>"
p11_login = "<boolean>"
start_gui = "<boolean>"
```

Príklad

```
[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"
```

Naimportovať objekty PKCS #11 zo súboru

Naimportuje objekty PKCS #11, ktoré boli vytvorené v súbore exportu PKCS #11.

Vyžadované atribúty

```
action_name = "IMPORT_OBJECTS"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
p11_object_file = "<názov súboru>"
```

Voliteľné atribúty

```
p11_login = "<boolean>" # REQUIRED TO IMPORT ANY PRIVATE OBJECTS
start_gui = "<boolean>"
```

Príklad

```
[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"
```

Vytvoriť samopodpísaný certifikát

Vytvorí samopodpísaný certifikát X.509 a priradené objekty PKCS #11 na symbole PKCS #11.

Vyžadované atribúty

```
action_name = "CREATE_SSC"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
p11_login = "TRUE"
p11_ssc_label = "<reťazec>"
p11_ssc_config = "<konfiguračný súbor openssl>"
```

Pričom <*konfiguračný súbor openssl*> predstavuje úplnú cestu v systéme UNIX ku konfiguračnému súboru OpenSSL, ktorý sa vyplní hodnotami použitými pri vytváraní samopodpísaného certifikátu.

Voliteľné atribúty

```
start_gui = "<boolean>"
```

Príklad

```
[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"
```

Vytvoriť požiadavku na podpísanie certifikátu PKCS #10

Vytvorí požiadavku na certifikáciu PKCS #10 alebo požiadavku na podpísanie certifikátu (CSR).

Vyžadované atribúty

```
action_name = "CREATE_CSR"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
p11_login = "TRUE"
p11_csr_label = "<reťazec>"
p11_csr_file = "<cesta k výstupnému súboru CSR>"
p11_csr_type = "<DER alebo Base64>'
p11_csr_config = "<konfiguračný súbor openssl>"
```

Kde <*DER alebo Base64*> vygeneruje výstupný súbor CSR zakódovaný algoritmom ASN.1 (DER) alebo výstupný súbor CSR zakódovaný algoritmom Base64 a <*cesta k výstupnému súboru CSR*> predstavuje úplnú cestu v systéme UNIX k výstupnému súboru CSR.

Voliteľné atribúty

```
start_gui = "<boolean>"
```

Príklad

```
[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"
```

V nástroji PKCS #11 Administration sú k dispozícii nasledujúce dávkové príkazy (p11admin).

Poznámka: Ak chcete použiť dávkové príkazy, postupujte takto:

1. Podľa postupu, popísaného v “Dávkové spracovanie” na strane 176, vytvorte a upravte dávkový súbor.
2. Vytvorte nové sekcie p11km_cmd, obsahujúce atribúty pre dávkové príkazy, ktoré chcete použiť.

Vypísať zoznam dostupných symbolov PKCS #11

Vygeneruje správu a zobrazí informácie o symboloch a slotoch pre dostupné symboly PKCS #11.

Vyžadované atribúty

```
action_name = "ADM_LIST_TOKENS"
```

Voliteľné atribúty

```
start_gui = "<boolean>"
```

Kde <*boolean*> je buď TRUE alebo FALSE

Príklad

```
[p11admin_cmd_list_tokens]
action_name = "ADM_LIST_TOKENS"
```

Vypísať zoznam dostupných mechanizmov PKCS#11

Vygeneruje správu a zobrazí dostupné mechanizmy PKCS #11, ktoré sú podporované symbolom PKCS #11 (identifikované na základe hodnôt atribútov ovládača a slotu).

Vyžadované atribúty

```
action_name = "ADM_LIST_MECHANISMS"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
```

Kde <*číslo slotu*> je hodnota kladného celého čísla a <*názov ovládača*> je niektorá z nasledujúcich hodnôt:

Hodnota	Description
AIX	Rámec Cryptographic Framework operačného systému AIX
IBM_4758_4960	Adaptéry šifrovacieho hardvéru IBM 4758/4960
IBM_4764	Adaptéry šifrovacieho hardvéru IBM 4764
Iné	Ak zadáte OTHER, musíte zadať aj atribút **p11_driver_path**.

Voliteľné atribúty

```
start_gui = "<boolean>"
```

Doplnkové atribúty

```
p11_driver_path = "<cesta k ovládaču PKCS#11>"
```

Pričom <*cesta k ovládaču PKCS#11*> predstavuje úplnú cestu v systéme UNIX k súboru knižnice PKCS #11, ktorá sa použije pri vykonaní príkazu. Tento atribút je možné zadať len v prípade, ak je atribút **p11_driver** nastavený na OTHER.

Príklad

```
[p11admin_cmd_list_4764_slot_0_mechs]
action_name = "ADM_LIST_MECHANISMS"
p11_driver = "IBM_4764"
p11_slot = "0"
start_gui = "TRUE"
```

Zobraziť informácie pre symbol PKCS #11

Zobrazí informácie o symboloch a slotoch PKCS #11 pre symbol PKCS #11.

Vyžadované atribúty

```
action_name = "ADM_SHOW_TOKEN_INFO"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
```

Voliteľné atribúty

```
start_gui = "<boolean>"
```

Príklad

```
[p11admin_cmd]
action_name = "ADM_SHOW_TOKEN_INFO"
p11_slot = "411"
p11_driver = "IBM_4764"
```

Inicializovať symbol PKCS #11:

Vykoná inicializáciu symbolu PKCS #11. Inicializácia zresetuje symbol, vymaže všetky uložené objekty a údaje PKCS#11 a umožní opätovné označenie symbolu.

Upozornenie: Keďže všetky objekty a údaje PKCS #11 sú počas procesu inicializácie vymazané, pred inicializáciou symbolu PKCS #11 skontrolujte, či tieto objekty a údaje nepotrebujete.

Vyžadované atribúty

```
action_name = "ADM_INIT_TOKEN"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>" ## SAME AS 'p11_init_slot'
p11_init_slot = "<číslo slotu>" ## SAME AS 'p11_slot'
p11_init_label = "<reťazec>" ## NEW TOKEN LABEL
```

Voliteľné atribúty

```
start_gui = "<boolean>"
```

Príklad

```
[p11admin_cmd]
action_name = "ADM_INIT_TOKEN"
p11_slot = "1"
p11_driver = "IBM_4764"
p11_init_slot = "1"
p11_init_label = "ABC Token"
```

Zobraziť hodiny pre symbol PKCS #11

Zobrazí hardvérové hodiny pre symbol PKCS #11, ak tento symbol má hodiny.

Vyžadované atribúty

```
action_name = "ADM_CLOCK_VIEW"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
```

Voliteľné atribúty

```
start_gui = "<boolean>"
```

Príklad

```
[p11admin_cmd]
action_name = "ADM_CLOCK_VIEW"
p11_slot = "1"
p11_driver = "IBM_4764"
```

Nastaviť hodiny pre symbol PKCS #11

Nastaví hardvérové hodiny pre symbol PKCS #11, ak tento symbol má hodiny.

Vyžadované atribúty

```
action_name = "ADM_CLOCK_SET"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
p11_clock_set = "<clock data>"
```

Kde <*clock data*> je aktuálny dátum a čas UTC v tomto formáte: HH:MM:SS mm-dd-YYYY.

Voliteľné atribúty

```
start_gui = "<boolean>"
```

Príklad

```
[p11admin_cmd]
action_name = "ADM_CLOCK_SET"
p11_slot = "1"
p11_driver = "IBM_4764"
p11_clock_set = "23:59:59 12-31-1999"
```

Resetovať PIN užívateľa symbolu PKCS #11

Zresetuje PIN užívateľa symbolu PKCS #11.

Vyžadované atribúty

```
action_name = "ADM_RESET_USER_PIN"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
```

Voliteľné atribúty

```
start_gui = "<boolean>"
```

Príklad

```
[p11admin_cmd_change_so_pin]
action_name = "ADM_RESET_USER_PIN"
p11_driver = "AIX"
p11_slot = "0"
```

Zmeniť PIN správcu bezpečnosti symbolu PKCS #11

Zmení PIN správcu bezpečnosti symbolu PKCS #11. Tento PIN sa používa pri vykonávaní administrácie symbolu.

Vyžadované atribúty

```
action_name = "ADM_CHANGE_SO_PIN"
p11_driver = "<názov ovládača>"
p11_slot = "<číslo slotu>"
```

Voliteľné atribúty

```
start_gui = "<boolean>"
```

Príklad

```
[p11admin_cmd_change_so_pin]
action_name = "ADM_CHANGE_SO_PIN"
p11_slot = "888"
p11_driver = "IBM_4764"
```

Pripojiteľné autentifikačné moduly

Rámec pripojiteľných autentifikačných modulov (PAM) poskytuje správcom systému schopnosť zapracovať do existujúceho systému viaceré autentifikačné mechanizmy prostredníctvom použitia pripojiteľných modulov.

Aplikácie s aktivovanou možnosťou využitia modulu PAM možno *pripojiť* do nových technológií bez modifikácie existujúcich aplikácií. Táto flexibilná aplikácia poskytuje administrátorom nasledovné možnosti:

- Vybrať v systéme ľubovoľnú službu autentifikácie pre aplikáciu
- Použiť pre danú službu viacero mechanizmov autentifikácie
- Pridať nové moduly autentifikačných služieb bez úprav v existujúcich aplikáciách
- Použiť predtým zadané heslo pre autentifikáciu s viacerými modulmi

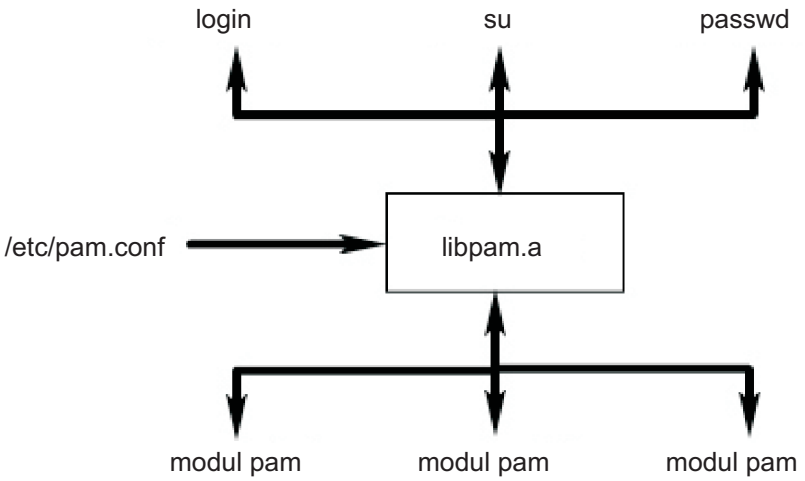
Rámec modulu PAM obsahuje knižnicu, pripojiteľné moduly a konfiguračný súbor. Knižnica modulu PAM slúži na implementáciu rozhrania API (Application programming interface), používa sa na riadenie transakcií modulu PAM a vyvolanie rozhrania SPI (Service programming interface) definovaného v pripojiteľných moduloch. Pripojiteľné moduly sa dynamicky načítajú do knižnice na základe vyvolania služby a príslušnej položky v konfiguračnom súbore. Úspech závisí nielen od pripojiteľného modulu, ale aj od správania definovaného pre službu. Pomocou koncepcie *triedenia do zásobníka* možno službu nakonfigurovať tak, aby sa na autentifikáciu používali viaceré spôsoby autentifikácie. V prípade podpory možno moduly nakonfigurovať na používanie skôr zadaného hesla namiesto zobrazenia výzvy na zadanie ďalších vstupných údajov.

Administrátor systému môže nakonfigurovať systém AIX tak, aby sa používalo PAM, úpravou atribútu **auth_type** v odseku usw súboru /etc/security/login.cfg. Nastavenie auth_type = PAM_AUTH nakonfiguruje, aby príkazy podporujúce PAM mohli volať rozhranie PAM API na účely autentifikácie priamo, miesto použitia historických autentifikačných rutín AIX. Táto konfigurácia je rozhodnutím o čase vykonania a nevyžaduje opätovné zavedenie, aby systém nadobudol účinnosť. Bližšie informácie o atribúte **auth_type**, nájdete v referencii súboru /etc/security/login.cfg. Nasledujúce natívne príkazy a aplikácie systému AIX boli upravené tak, aby rozpoznávali atribút **auth_type** a podporovali autentifikáciu PAM:

- **login**
- **passwd**
- **su**
- **ftp**
- **telnet**
- **rlogin**
- **rexec**
- **rsh**
- **snappd**
- **imapd**

- **dtaction**
- **dtlogin**
- **dtsession**

Nasledujúce obrázky znázorňujú interakciu medzi aplikáciami povoľujúcimi PAM, knižnicou PAM, konfiguračným súborom a modulmi PAM na systéme, ktorý bol nakonfigurovaný na použitie PAM. Aplikácie s povolenou autentifikačnou metódou PAM vyvolávajú v knižnici PAM rozhranie PAM API. Knižnica určí príslušný modul na načítanie na základe položky aplikácie v konfiguračnom súbore a vyvolá v danom module rozhranie PAM SPI. Komunikácia medzi modulom PAM a aplikáciou sa uskutočňuje prostredníctvom funkcie konverzácie implementovanej v aplikácii. Úspech alebo zlyhanie komunikácie s modulom a akcie definované v konfiguračnom súbore potom určia, či je potrebné načítať ďalší modul. Ak áno, proces bude pokračovať; v opačnom prípade sa výsledok odovzdá späť aplikácii.



Obrázok 3. Aplikačný rámec modulu PAM a entity. Na tejto ilustrácii je znázornené, ako príkazy s povoleným PAM používajú knižnicu PAM na prístup k príslušnému modulu PAM.

Knižnica PAM

PAM library,/usr/lib/libpam.a obsahuje API PAM, ktoré slúži ako spoločné rozhranie pre všetky aplikácie PAM a tiež riadi zavádzanie modulu.

Knižnica PAM uskutoční načítanie modulov na základe koncepcie triedenia do zásobníka definovanej v súbore /etc/pam.conf.

Nasledovné funkcie rozhrania PAM API vyvolajú príslušné rozhranie PAM SPI dostupné v module PAM. Napríklad, API pam_authenticate vyvolá SPI pam_sm_authenticate v module PAM.

- pam_authenticate
- pam_setcred
- pam_acct_mgmt
- pam_open_session
- pam_close_session
- pam_chauthtok

Knižnica PAM zahŕňa aj niekoľko rámcových rozhraní API umožňujúcich aplikácii vyvolať moduly PAM a odovzdať im informácie. Nasledujúca tabuľka zobrazuje rámcové API modulov PAM implementovaných v AIX a ich funkcie:

Rámcové rozhrania API modulov PAM	Funkcia
pam_start	Vytvorí reláciu s modulom PAM
pam_end	Ukončí reláciu s modulom PAM
pam_get_data	Vyhľadá údaje špecifické pre modul
pam_set_data	Nastaví údaje špecifické pre modul
pam_getenv	Načíta hodnotu definovanej premennej prostredia PAM
pam_getenvlist	Načíta zoznam všetkých definovaných premenných prostredia PAM a ich hodnoty
pam_putenv	Nastaví premennú prostredia PAM
pam_get_item	Vyhľadá spoločné informácie modulov PAM
pam_set_item	Nastaví spoločné informácie modulov PAM
pam_get_user	Vyhľadá meno užívateľa
pam_strerror	Načíta štandardné chybové hlásenie pre modul PAM

Moduly PAM

Moduly PAM umožňujú kolektívne alebo nezávislé použitie viacerých mechanizmov autentifikácie v systéme.

Daný modul PAM musí implementovať aspoň jeden zo štyroch typov modulov. V ďalšom sú opísané typy modulov spolu so zodpovedajúcimi PAM SPI, u ktorých sa vyžaduje, aby vyhovovali typu modulu.

Moduly autentifikácie
Vykonávajú autentifikáciu užívateľov a nastavenie, obnovenie alebo odstránenie oprávnení. Tieto moduly identifikujú užívateľa na základe autentifikácie a oprávnení.

Funkcie modulov autentifikácie:

- pam_sm_authenticate
- pam_sm_setcred

Moduly manažmentu kont
Určujú platnosť konta užívateľa a ďalší prístup po identifikácii pomocou modulu autentifikácie. Kontroly uskutočnené týmito modulmi zvyčajne zahŕňajú kontrolu ukončenia platnosti konta a obmedzení hesla.

Funkcia modulu manažmentu kont:

- pam_sm_acct_mgmt

Moduly riadenia relácií
Spúšťajú a ukončujú užívateľské relácie. Okrem toho môže byť poskytnutá podpora pre auditovanie relácií.

Funkcie modulov riadenia relácií:

- pam_sm_open_session
- pam_sm_close_session

Moduly na správu hesla
Vykonávajú modifikáciu hesla a súvisiacu správu atribútov.

Funkcie modulov na správu hesla:

- pam_sm_chauthtok

Konfiguračný súbor PAM

Konfiguračný súbor /etc/pam.conf obsahuje položky služieb pre každý typ modulu PAM a slúži na smerovanie služieb cestou definovanou pre modul.

Položky v súbore pozostávajú z nasledovných polí oddelených medzerami:

```
service_name module_type control_flag module_path module_options
```

Nasledujú popisy týchto polí:

service_name
Určuje názov služby. Kľúčové slovo OTHER sa používa na definovanie predvoleného modulu používaného s aplikáciami, ktoré nie sú určené v niektorej položke.

module_type
Určuje typ modulu pre službu. Platnými typmi modulu sú **auth**, **account**, **session** alebo **password**. Daný modul poskytne podporu pre jeden alebo viacero typov modulov.

control_flag
Určuje akcie triedenia modulov do zásobníka. Podporované príznaky riadenia sú požadované, nevyhnutné, záväzné alebo voliteľné.

module_path

Zadáva modul, ktorý sa má zaviesť pre službu. Platné hodnoty pre *module_path* môžu byť uvedené ako úplná cesta k modulu alebo iba ako názov modulu. Ak je uvedená úplná cesta k modulu, knižnica PAM použije *module_path* na zavedenie 32-bitových služieb a podadresár 64 na zavedenie 64-bitových služieb. Ak neuvediete úplnú cestu k modulu, knižnica PAM pripojí k názvu modulu predponu /usr/lib/security (v prípade 32-bitových služieb) alebo /usr/lib/security/64 (v prípade 64-bitových služieb).

module_options
Uvádza medzerami oddelený zoznam volieb, ktoré možno odovzdať modulom služieb. Hodnoty v tomto poli závisia od volieb podporovaných modulom, ktorý je definovaný v poli *module_path*. Toto pole je voliteľné.

Poškodené položky alebo položky s nesprávnymi hodnotami pre polia **module_type** alebo **control_flag** sú knižnicou PAM ignorované. Položky začínajúce znakom čísla (#) na začiatku riadka sú takisto ignorované, pretože tento znak predstavuje poznámku.

PAM podporuje koncept, ktorý sa zvyčajne nazýva "vrstvenie" a ktorý umožňuje pre každú službu použiť viaceré mechanizmy. Vrstvenie sa do konfiguračného súboru implementuje vytvorením viacerých položiek pre službu s rovnakým poľom **module_type**. Moduly sa vyvolávajú v poradí, v ktorom sú uvedené v súbore pre danú službu, pričom konečný výsledok je určený poľom **control_flag** zadaným pre každú položku. Platné hodnoty pre pole **control_flag** a príslušné akcie pre zásobník sú nasledovné:

Hodnota poľa control_flag	Správanie
required	Načítanie všetkých požadovaných modulov v zásobníku musí byť úspešne dokončené. Ak jeden alebo viacero požadovaných modulov zlyhá, uskutoční sa opätovný pokus o načítanie všetkých požadovaných modulov v zásobníku, ale pri prvom zlyhaní načítania požadovaného modulu sa vráti chyba.
requisite	Podobá sa vyžadovanému s tou výnimkou, že ak nevyhnutný modul zlyhá, nebudú sa spracovávať žiadne ďalšie moduly v zásobníku a prvý kód zlyhania bude ihneď vrátený z požadovaného alebo nevyhnutného modulu.
sufficient	Ak je modul, ktorý je označený príznakom ako záväzný, úspešne načítaný a žiadne predchádzajúce požadované alebo záväzné moduly nezlyhali, všetky zostávajúce moduly v zásobníku sú ignorované a vráti sa výsledok úspešného spracovania.
optional	Ak v zásobníku nie sú požadované moduly a žiadny záväzný modul nebol úspešne spracovaný, je potrebné úspešne načítať aspoň jeden modul pre danú službu. Ak sa úspešne načíta iný modul zo zásobníka, zlyhanie voliteľného modulu sa ignoruje.

Nasledujúca podsada /etc/pam.conf je príkladom vrstvenia v type modulu auth pre službu prihlásenia.

```
#
# PAM configuration file /etc/pam.conf
#

# Authentication Management
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
```

Príklad konfiguračného súboru obsahuje tri položky pre službu prihlásenia. Ak je podľa potreby zadané pam_ckfile i pam_aix, budú spustené oba moduly a oba musia byť úspešné, aby bol úspešný celkový výsledok. Tretia položka pre

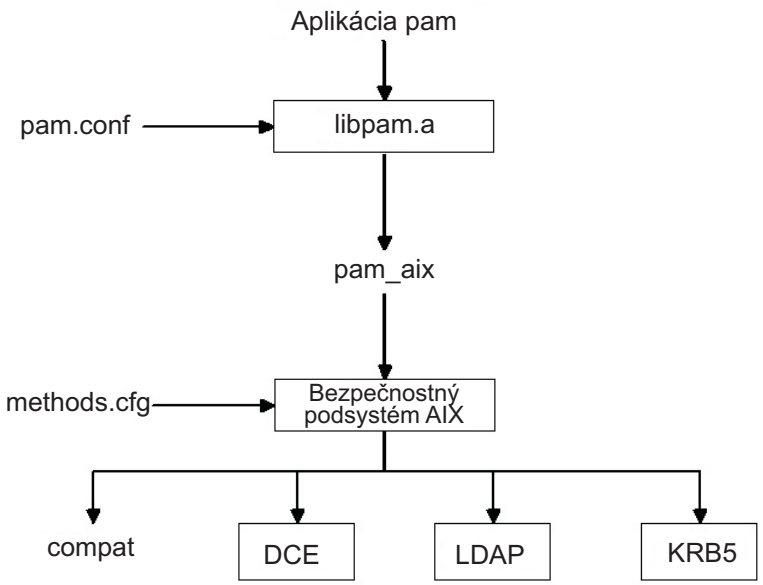
fiktívny modul pam_test je voliteľná a jej úspech alebo neúspech nebude mať vplyv na skutočnosť, či sa užívateľ môže prihlásiť. Voľba use_first_pass modulu pam_test vyžaduje, aby sa namiesto výzvy na zadanie nového hesla použilo predtým zadané heslo.

Použitie kľúčového slova OTHER ako názvu služby umožní štandardné nastavenie pre všetky ostatné služby, ktoré nie sú explicitne deklarované v konfiguračnom súbore. Nastavením predvoleného modulu sa zaistí, že vo všetkých prípadoch existuje pre daný typ modulu najmenej jeden modul. V tomto príklade všetky služby s výnimkou prihlásenia vždy zlyhajú, keďže modul pam_prohibit vráti zlyhanie PAM pre všetky vyvolania.

Modul pam_aix

Modul pam_aix je modul PAM, ktorý poskytuje aplikáciám podporujúcim PAM prístup k službám zabezpečenia AIX poskytovaním rozhraní, ktoré volajú ekvivalentné služby AIX, ak existujú.

Tieto služby sú vykonávané zavádzateľným modulom autentifikácie alebo vstavanou funkciou AIX založenou na užívateľovej definícii a príslušnom nastavení v súbore methods.cfg. Kódy chýb generované počas vykonania služby systému AIX sa mapujú k príslušným kódom chýb modulu PAM.



Obrázok 4. Cesta volania aplikácie PAM do subsystému zabezpečenia AIX

Tento obrázok zobrazuje cestu volania API aplikácie PAM, v prípade, že súbor /etc/pam.conf je nakonfigurovaný na používanie modulu pam_aix. Ako ukazuje diagram, integrácia umožňuje autentifikáciu užívateľov pomocou ktorýchkoľvek zavádzateľných modulov (DCE, LDAP alebo KRB5) alebo v súboroch AIX (compat).

Modul pam_aix je nainštalovaný v adresári /usr/lib/security. Integrácia modulu pam_aix vyžaduje, aby bol súbor /etc/pam.conf nakonfigurovaný na používanie tohto modulu. Ukladanie do zásobníka je stále dostupné, ale nie je zobrazené v nasledujúcom príklade súboru /etc/pam.conf:

```
#
# Authentication management
#
OTHER   auth     required       /usr/lib/security/pam_aix

#
# Account management
#
OTHER   account  required       /usr/lib/security/pam_aix

#
# Session management
```

```
#
OTHER   session  required       /usr/lib/security/pam_aix

#
# Password management
#
OTHER   password required       /usr/lib/security/pam_aix
```

Modul pam_aix má implementácie pre SPI funkcie pam_sm_authenticate, pam_sm_chauthok a pam_sm_acct_mgmt. SPI pam_sm_setcred, pam_sm_open_session a pam_sm_close_session sú tiež implementované v module pam_aix, ale tieto funkcie SPI vracajú vyvolania PAM_SUCCESS.

Nasleduje aproximatívne mapovanie volaní PAM SPI do zabezpečovacieho podsystému AIX:

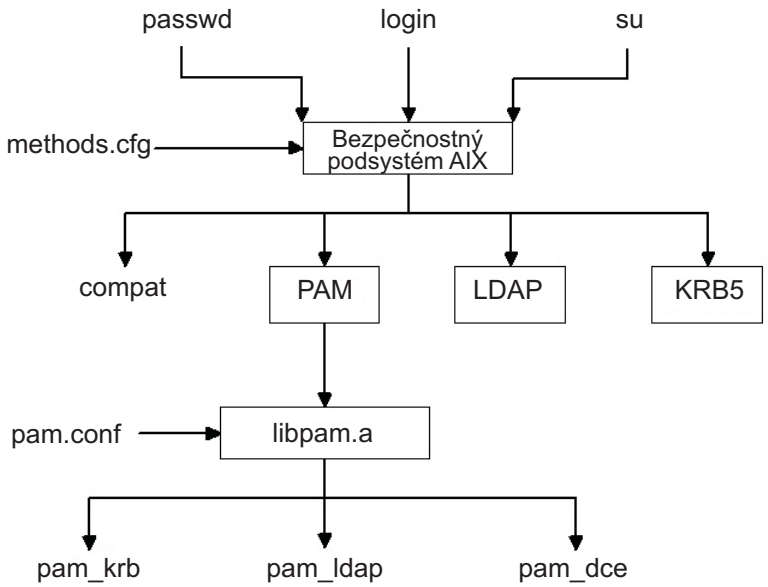
```
 PAM SPI                    AIX
=========                  =====
pam_sm_authenticate   -->  authenticate
pam_sm_chauthtok      -->  passwdexpired, chpass
                           Note: passwdexpired is only checked if the
                           PAM_CHANGE_EXPIRED_AUTHTOK flag is passed in.
pam_sm_acct_mgmt      -->  loginrestrictions, passwdexpired
pam_sm_setcred        -->  No comparable mapping exists, PAM_SUCCESS returned
pam_sm_open_session   -->  No comparable mapping exists, PAM_SUCCESS returned
pam_sm_close_session  -->  No comparable mapping exists, PAM_SUCCESS returned
```

Údaje, ktoré je plánované poskytnúť zabezpečovaciemu podsystému AIX, je možné nastaviť použitím funkcie pam_set_item, prednostne pred použitím modulu, alebo použitím modulu pre údaje pam_aix, ak ešte neexistuje.

Zavádzateľný modul autentifikácie PAM

Služby zabezpečenia AIX môžu byť nakonfigurované na volanie modulov PAM použitím existujúceho systému zavádzateľných modulov autentifikácie AIX.

Ak je súbor /usr/lib/security/methods.cfg nastavený správne, zavádzací modul PAM smeruje služby zabezpečenia AIX (passwd, login a tak ďalej) do knižnice PAM. Knižnica PAM kontroluje súbor /etc/pam.conf na zistenie, ktorý modul PAM sa má použiť a následne vykoná príslušné volanie PAM SPI. Návratové hodnoty z modulu PAM sa mapujú ku kódom chýb systému AIX a sú odoslané späť do volajúceho programu.
Tento obrázok uvádza cestu, ktorú volanie zabezpečovacích služieb AIX použije, ak je konfigurácia PAM správna.



Obrázok 5. Cesta volania modulu PAM zabezpečovacou službou systému AIX

Ukázané moduly PAM (pam_krb, pam_ldap a pam_dce) sú uvedené ako príklady riešení tretích strán.

Zavádzací modul PAM je nainštalovaný v adresári /usr/lib/security a je modulom určeným len pre autentifikáciu. Modul PAM je nutné prepojiť s databázou, aby vznikol zložený zavádzací modul. V ďalšom príklade sú zobrazené sekcie, ktoré je možné pridať do súboru methods.cfg, aby vznikol zložený modul PAM s databázou nazývanou databáza súborov. Kľúčové slovo BUILTIN pre atribút **db** určuje databázu ako súbory systému UNIX.

```
PAM:
        program = /usr/lib/security/PAM

PAMfiles:
        options = auth=PAM,db=BUILTIN
```

Vytvorenie a modifikácia užívateľov sa potom vykoná s použitím voľby -R s príkazmi administrácie a nastavením atribútu *SYSTEM*, keď bude užívateľ vytvorený. Napríklad:

```
mkuser -R PAMfiles SYSTEM=PAMfiles registry=PAMfiles pamuser
```

Táto akcia informuje nasledujúce volania zabezpečovacích služieb AIX (login, passwd, atď.) o tom, že sa má použiť pre autentifikáciu zavádzací modul PAM. Napriek tomu, že bola pre zložený modul v tomto príklade použitá databáza **súborov**, môžu byť tiež použité aj iné databázy, ako je LDAP, ak sú nainštalované. Výsledkom vytvorenia užívateľov podľa vyššie uvedeného popisu je nasledovné mapovanie zabezpečenia systému AIX k volaniam rozhrania PAM API:

```
 AIX                     PAM API
=====                   =========
authenticate       -->  pam_authenticate
chpass             -->  pam_chauthtok
passwdexpired      -->  pam_acct_mgmt
passwdrestrictions -->  No comparable mapping exists, success returned
```

Prispôsobením súboru /etc/pam.conf je možné nasmerovať volania rozhrania PAM API do požadovaného modulu PAM na autentifikáciu. Pre ďalšie zdokonalenie mechanizmu autentifikácie môže byť implementované vrstvenie.

Údaje požadované zabezpečovacou službou systému AIX sú postúpené do modulu PAM prostredníctvom funkcie pam_set_item, pretože užívateľské dialógové okno z modulu PAM nie je možné použiť. Moduly PAM, napísané pre integráciu s modulom PAM by mali získavať všetky údaje s volaniami pam_get_item a nemali by žiadať užívateľa o vstupné údaje, pretože toto je zabezpečené zabezpečovacou službou.

Zistenie cyklu sa poskytuje na zachytenie možných konfiguračných chýb, v ktorých je zabezpečovacia služba AIX smerovaná do PAM a potom sa modul PAM pokúsi o volanie zabezpečovacej služby AIX na vykonanie operácie. Zistenie tejto udalosti cyklu bude mať za následok okamžité zlyhanie zamýšľanej operácie.

Poznámka: Súbor /etc/pam.conf by *nemal* byť zapísaný pre použitie modulu pam_aix pri použití integrácie PAM zo služby zabezpečenia AIX pre modul PAM, pretože to vyvolá stav cyklu.

Pridanie modulu PAM

Ak chcete zapnúť viaceré mechanizmy autentifikácie, môžete pridať modul PAM.

1. Umiestnite 32-bitovú verziu modulu do adresára /usr/lib/security a 64-bitovú verziu modulu do adresára /usr/lib/security/64.
2. Nastavte vlastníctvo súboru na root a oprávnenia na 555. Knižnica PAM nezavedie žiaden modul, ktorý nie je vlastnený užívateľom s oprávneniami typu root.
3. Aktualizujte konfiguračný súbor /etc/pam.conf, aby bol modul zahrnutý v položkách pre požadované názvy služieb.
4. Otestujte príslušné služby, aby ste sa uistili, že sú funkčné. Neodhlasujte sa zo systému pred dokončením testu prihlásenia.

Zmena súboru /etc/pam.conf

Pred zmenou súboru /etc/pam.conf by ste mali zvážiť niekoľko vecí.

Keď budete meniť konfiguračný súbor /etc/pam.conf, zvážte nasledujúce požiadavky:

- Súbor má byť vždy vo vlastníctve kmeňového užívateľa a skupinovej bezpečnosti. Povolenia na súbor musia byť 644, aby mohol mať každý prístup na čítanie alebo len kmeňový užívateľ mohol vykonať modifikácie.
- Pre väčšiu bezpečnosť zvážte explicitnú konfiguráciu každej služby povoľujúcej PAM a následné použitie modulu pam_prohibit pre kľúčové slovo OTHER.
- Prečítajte si všetku dokumentáciu dodávanú pre vybratý modul a určte, ktoré príznaky riadenia a voľby sú podporované, a aký bude ich účinok.
- Pozorne si vyberte zoradenie modulov a príznakov riadenia a nezabudnite na správanie sa príznakov riadenia požadovaný, nevyhnutný, záväzný a voliteľný v moduloch v zásobníku.

Poznámka: Nesprávna konfigurácia konfiguračného súboru PAM môže viesť k tomu, že sa do systému nebude dať prihlásiť, pretože konfigurácia sa použije na všetkých užívateľov vrátane kmeňových užívateľov. Po vykonaní zmien súboru a pred odhlásením zo systému vždy otestujte príslušné aplikácie. V prípade, že do systému nie je možné sa prihlásiť, môžete ho obnoviť zavedením systému v režime údržby a opravou údajov v konfiguračnom súbore /etc/pam.conf.

Povolenie ladenia PAM

Knižnica PAM (Pluggable Authentication Modules) môže poskytovať informácie o ladení počas vykonávania. Po povolení výstupu z ladenia v systéme môžete pomocou zhromaždených informácií sledovať volania rozhrania PAM API a zistiť body zlyhania v aktuálnom nastavení PAM.

Výstup z ladenia PAM môžete povoliť vykonaním nasledujúcich krokov:

1. V adresári /etc/pam_debug vytvorte prázdny súbor s názvom pam_debug pomocou príkazu **touch**, ak tento súbor ešte neexistuje. Knižnica PAM skontroluje, či existuje súbor /etc/pam_debug, a povolí výstup syslog, ak sa nájde tento súbor.
2. Upravte súbor /etc/syslog.conf a určte v ňom súbor, do ktorého sa budú zapisovať správy auth syslog na požadovanej úrovni priority. Napríklad, ak chcete, aby sa správy úrovne ladenia PAM zapisovali do súboru /var/log/auth.log, pridajte nasledujúci text ako nový riadok v súbore syslog.conf:

   ```
   *.debug /var/log/auth.log
   ```
3. Zadaním príkazu **touch** vytvorte výstupný súbor, na ktorý sa odkazuje krok 2 (/var/log/auth.log), ak ešte neexistuje.
4. Ak chcete reštartovať démona syslogd, aby sa prejavili zmeny v konfigurácii, vykonajte nasledujúce kroky:
   a. Zastavte démona syslog zadaním príkazu:

      ```
      stopsrc -s syslogd
      ```
   b. Spustite démona syslog zadaním príkazu:

      ```
      startsrc -s syslogd
      ```

Po spustení aplikácie PAM sa budú správy z ladenia zhromažďovať do výstupného súboru definovaného v konfiguračnom súbore /etc/syslog.conf.

Podpora OpenSSH a Kerberos verzie 5

Kerberos je autentifikačný mechanizmus, ktorý poskytuje zabezpečenie autentifikácie pre sieťových užívateľov. Zabraňuje prenosu textových hesiel cez sieť zakódovaním autentifikačných správ medzi klientmi a servermi. Okrem toho, Kerberos poskytuje systém pre autorizáciu vo forme tokenov administrácie alebo oprávnení.

Na autentifikáciu užívateľa použitím Kerberos, užívateľ spustí príkaz **kinit** na získanie úvodných oprávnení z centrálneho servera Kerberos, známeho ako KDC (Key Distribution Center). KDC overí užívateľa a vráti mu jeho úvodné oprávnenia, známe ako TGT (Ticket-Granting Ticket). Užívateľ môže potom spustiť reláciu vzdialeného prihlásenia napríklad pomocou služby Telnet povolenej pomocou Kerberos alebo pomocou OpenSSH a Kerberos autentifikuje užívateľa získaním splnomocnení užívateľa z KDC. Kerberos vykoná túto autentifikáciu bez zásahu užívateľa, takže užívatelia nemusia zadávať prihlasovacie heslá. Verzia Kerberos od IBM je známa ako NAS (Network Authentication Service). NAS možno nainštalovať z CD rozširujúceho balíka AIX, kde je dostupný v balíkoch krb5.client.rte a krb5.server.rte. Počnúc vydaním OpenSSH 3.6 z júla 2003 OpenSSH podporuje autentifikáciu a autorizáciu Kerberos 5 prostredníctvom NAS verzie 1.3.

OpenSSH Verzia 3.8 a novšie podporujú autentifikáciu Kerberos 5 a autorizáciu cez NAS Verzia 1.4. Migrácia z predchádzajúcich verzií NAS (Kerberos) sa musí uskutočniť pred aktualizáciou OpenSSH. OpenSSH Verzia 3.8.x bude fungovať len s NAS Verzia 1.4 alebo novšou.

AIX vytvoril OpenSSH s autentifikáciou Kerberos ako voliteľnou metódou. Ak knižnice Kerberos nie sú počas chodu OpenSSH nainštalované na systém, autentifikácia Kerberos bude vynechaná a OpenSSH skúsi ďalšiu nakonfigurovanú metódu autentifikácie (napríklad autentifikáciu AIX).

Po inštalácii Kerberos sa odporúča, aby ste si skôr, ako začnete konfigurovať server Kerberos, prečítali dokumentáciu. Viac informácií o spôsobe inštalácie a správy pre Kerberos nájdete v *IBM Network Authentication Service Version 1.3 for AIX : Administrator's and User's Guide* v /usr/lpp/krb5/doc/html/*lang*/ADMINGD.htm.

Súvisiace informácie:

OpenSSH

Obrazy OpenSSH

Použite nasledujúce kroky na inštaláciu obrazov OpenSSH:

1. Navštívte webovú lokalitu AIX Web Download Pack Programs.

   Poznámka: Obraz sofvéru OpenSSH sa poskytuje na médiu so základným operačným systémom AIX, no tento obraz sa predvolene nenainštaluje.
2. Kliknite na možnosť **Downloads** v časti Additional information.
3. Prihláste sa s použitím vášho ID a hesla, aby ste získali prístup k dostupným balíkom.
4. Vyberte možnosť **OpenSSH** a kliknite na tlačidlo **Continue**.
5. Akceptujte licenčnú zmluvu a stiahnite si balík.
6. Rozbaľte balík s obrazom zadaním príkazu **uncompress** *názov_balíka*. Napríklad:
   ```
   uncompress OpenSSH_6.0.0.6203.tar.Z
   ```
7. Balík rozbaľte príkazom **tar -xvf** *packagename*. Napríklad:
   ```
   tar -xvf OpenSSH_6.0.0.6203.tar
   ```
8. Spustite príkaz **inutoc**.
9. Spustite príkaz **smitty install**.
10. Vyberte **Install and Update Software**.
11. Vyberte **Update Installed Software to Latest Level (Update All)**.
12. V poli **INPUT device / directory for software** napíšte bodku (.) a stlačte kláves Enter.
13. Rolujte nadol na **ACCEPT new license agreements** a stlačte kláves **Tab** pre zmenu poľa na **Yes**.
14. Spustite inštaláciu dvojitým stlačením klávesu Enter.

Obrazy OpenSSH sú obrazy základnej úrovne, nie dočasné opravy programu (PTF). Pri inštalácii sa všetok predchádzajúci kód predchádzajúcej verzie prepíše obrazmi novej verzie.

Konfigurácia kompilácie OpenSSH

Nasledujúce informácie uvádzajú, ako sa kompiluje kód OpenSSH pre AIX.

Pri konfigurácii OpenSSH pre AIX verzie 6.1 je výstup podobný tomuto:

```
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
```

Poznámka: Voľby kompilácie pre AIX verzie 6.1 a AIX, verzia 7.1 sú podobné, pretože binárnosti oboch verzií sú rovnaké.

Používanie OpenSSH s Kerberos

Na použitie OpenSSH s Kerberos sa vyžaduje úvodné nastavenie.

Nasledujúci postup poskytuje informácie o úvodnom nastavení, ktoré sa vyžaduje na používanie OpenSSH s Kerberos:

1. Na vašich klientoch a serveroch OpenSSH musí existovať súbor /etc/krb5.conf. Tento súbor informuje Kerberos, ktorý KDC sa má použiť, aká životnosť má byť pridelená každému lístku, atď. Nasleduje príklad súboru krb5.conf:

```
[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. Taktiež musíte pridať nasledujúce služby Kerberos každému súboru počítača klienta /etc/services:

```
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. Ak vaše KDC používa LDAP ako register pre ukladanie užívateľských informácií, prečítajte si “Zavádzací modul autentifikácie LDAP” na strane 146 a publikácie Kerberos. Okrem toho sa presvedčte, či boli vykonané nasledujúce akcie:
   - KDC spúšťa klienta LDAP. Môžete spustiť démona klienta LDAP príkazom **secldapclntd**.
   - LDAP server spúšťa démona servera LDAP slapd.
4. Na serveri OpenSSH zmeňte súbor /etc/ssh/sshd_config tak, aby obsahoval nasledujúce riadky:

   ```
   KerberosAuthentication yes
   KerberosTicketCleanup yes
   GSSAPIAuthentication yes
   GSSAPICleanupCredentials yes
   UseDNS yes
   ```

   Ak je UseDNS nastavené na **Yes**, server ssh vykoná spätné vyhľadanie hostiteľa, aby našiel názov pripájajúceho sa klienta. Je to nevyhnutné pri používaní autentifikácie na báze hostiteľa alebo keď chcete, aby sa v informáciách o poslednom prihlásení namiesto IP adries zobrazili názvy hostiteľov.

   Poznámka: Niektoré relácie ssh sa pri vykonávaní obráteného vyhľadávania mien zastavia, pretože servery DNS neprístupné. Ak nastane táto situácia, môžete preskočiť vyhľadávania DNS nastavením UseDNS na hodnotu no. Ak UseDNS nie je explicitne nastavené v súbore /etc/ssh/sshd_config, predvolená hodnota bude UseDNS yes.
5. Na serveri SSH spustite príkaz **startsrc -g ssh** na spustenie démona servera ssh.
6. Na počítači klienta SSH spustite príkaz **kinit** na získanie úvodných oprávnení (TGT). Spustením príkazu **klist** môžete overiť, či ste prijali TGT. Tento príkaz zobrazí všetky vaše oprávnenia.
7. Pripojte sa k serveru spustením príkazu **ssh** *meno_užívateľa@názov_servera*.
8. Ak je Kerberos správne nakonfigurovaný na autentifikáciu užívateľa výzva na zadanie hesla sa neobjaví a užívateľ bude automaticky prihlásený k serveru SSH.

Zabezpečenie siete

Nasledujúce časti opisujú inštaláciu a konfiguráciu bezpečnosti IP; identifikáciu potrebných a nepotrebných sieťových služieb a audit a monitorovanie zabezpečenia siete.

Zabezpečenie TCP/IP

Ak ste nainštalovali TCP/IP (Transmission Control Protocol/Internet Protocol) a softvér (Network File System), môžete svoj systém nakonfigurovať pre komunikáciu v sieti.

Táto príručka sa nezaoberá základmi protokolu TCP/IP, skôr je zameraná na otázky bezpečnosti v súvislosti s protokolom TCP/IP. Informácie o inštalácii úvodnej konfigurácii TCP/IP nájdete v časti Transmission Control Protocol/Internet Protocol v *Networks and communication management*.

Osoba spravujúca systém môže z rôznych dôvodov chcieť dosiahnuť určitú úroveň bezpečnosti. Požiadavka na úroveň bezpečnosti môže napríklad vyplývať z podnikovej politiky. Konkrétna úroveň bezpečnosti sa tiež môže požadovať pre zabezpečenie prístupu systému do vládnych systémov. Tieto bezpečnostné štandardy je možné aplikovať v sieti, operačnom systéme, aplikačnom softvéri, dokonca aj v programoch napísaných osobou spravujúcou váš systém.

Táto časť popisuje funkcie bezpečnosti, ktoré poskytuje TCP/IP v štandardnom režime a tiež ako zabezpečený systém a zaoberá sa niektorými hľadiskami bezpečnosti, ktoré prislúchajú sieťovému prostrediu.

Po inštalácii softvéru TCP/IP a NFS pomocou rýchlej cesty **tcpip** nástroja SMIT (System Management Interface Tool) nakonfigurujte svoj systém.

Ďalšie informácie o príkaze **dacinet** nájdete v časti *Commands Reference*.

Špecifické zabezpečenie operačných systémov

Mnohé funkcie bezpečnosti, ako napríklad riadenie prístupu do siete a auditovanie sietí, ktoré sú dostupné pre TCP/IP sú založené na dostupných funkciách v operačných systémoch.

Bezpečnosť TCP/IP je načrtnutá v ďalších častiach.

Riadenie prístupu do siete:

Bezpečnostná politika pre prácu v sieti je rozšírením bezpečnostnej politiky pre operačný systém a skladá sa z autentifikácie užívateľov, autentifikácie pripojení a bezpečnosti údajov.

Skladá sa z nasledujúcich hlavných komponentov:

- Autentifikácia užívateľa sa vykonáva na vzdialenom hostiteľovi podľa mena užívateľa a hesla rovnakým spôsobom, ako keď sa užívateľ prihlási do lokálneho systému. Dôveryhodné príkazy TCP/IP, ako sú napríklad **ftp**, **rexec** a **telnet**, majú rovnaké požiadavky a podliehajú rovnakému overovaciemu procesu ako dôveryhodné príkazy v operačnom systéme.
- Autentifikácia pripojenia overuje, či má vzdialený hostiteľ očakávanú adresu Internet Protocol (IP) a názov. Týmto sa vzdialenému hostiteľovi zabráni v maskovaní sa a vydávaní sa za iného vzdialeného hostiteľa.
- Zabezpečenie importu a exportu údajov povoľuje údajom na určitej úrovni bezpečnosti tiecť do a z adaptérov sieťového rozhrania s rovnakou úrovňou bezpečnosti a úrovňami oprávnení. Napríklad, prísne utajované údaje môžu tiecť len medzi adaptérmi, ktoré sú nastavené na príslušnú úroveň bezpečnosti pre prenos takýchto údajov.

Auditovanie siete:

Auditovanie siete zabezpečuje TCP/IP, s využitím podsystému auditu, ktorý audituje aplikačné programy.

Úlohou auditovania je zaznamenávať tie akcie, ktoré majú vplyv na bezpečnosť systému, a informácie o užívateľovi, ktorý je za takéto akcie zodpovedný.

Auditované sú nasledujúce udalosti aplikácií:

- Prístup do siete
- Pripojenie
- Export údajov
- Import údajov

Vytváranie a odstraňovanie objektov audituje operačný systém. Záznamy auditu aplikácií pozastavujú a obnovujú auditovanie, čím sa predchádza nadbytočnému auditovaniu vykonávanému jadrom.

Dôveryhodná cesta, dôveryhodné prostredie Shell a Secure Attention Key:

Operačný systém poskytuje *dôveryhodnú cestu* za účelom zabránenia neautorizovaným programom v čítaní údajov na termináli užívateľa. Táto cesta sa používa, keď sa pre systém vyžaduje zabezpečená komunikačná cesta, ako napríklad, pri zmene hesiel alebo pri prihlasovaní sa do systému.

Operačný systém tiež poskytuje *trusted shell* (**tsh**), ktorý spúšťa len dôveryhodné programy otestované a overené ako bezpečné. Protokol TCP/IP podporuje obe tieto funkcie spolu s funkciou *secure attention key* (SAK), ktorá vytvára prostredie potrebné pre bezpečnú komunikáciu medzi vami a systémom. Lokálna funkcia SAK je k dispozícii vždy, keď používate protokol TCP/IP. Vzdialená funkcia SAK je k dispozícii prostredníctvom príkazu **telnet**.

Lokálna funkcia SAK má v **telnete** rovnakú funkciu, akú má v ostatných aplikačných programoch operačného systému: ukončí proces **telnet** a všetky ďalšie procesy súvisiace s terminálom, v ktorom bol **telnet** spustený. V programe telnet však môžete odoslať požiadavku na dôveryhodnú cestu do vzdialeného systému, keď použijete príkaz **telnet send sak** (kým budete v príkazovom režime **telnet**). Na inicializáciu požiadavky SAK môžete tiež definovať jeden kláves, a to použitím príkazu **telnet set sak**.

Ďalšie informácie o súčasti Trusted Computing Base obsahuje “Trusted Computing Base” na strane 1.

Bezpečnosť príkazov TCP/IP

Niektoré príkazy v protokole TCP/IP poskytujú počas prevádzky zabezpečené prostredie. Týmito príkazmi sú **ftp**, **rexec** a **telnet**.

Funkcia **ftp** poskytuje zabezpečenie počas prenosu súborov. Príkaz **rexec** poskytuje zabezpečené prostredie pre spúšťanie príkazov na cudzom hostiteľovi. Funkcia **telnet** poskytuje zabezpečenie pre prihlasovanie sa k cudziemu hostiteľovi.

Príkazy **ftp**, **rexec** a **telnet** poskytujú zabezpečenie len počas ich vykonávania. To znamená, že nenastavia zabezpečené prostredie, ktoré by mohli používať iné príkazy. Ak chcete zabezpečiť systém aj pre ďalšie operácie, použite príkaz **securetcpip**. Tento príkaz vám umožní zabezpečiť systém tým, že zakáže nedôveryhodné aplikácie a démonov a poskytne vám tiež možnosť zabezpečiť sieťový protokol vrstvy IP.

Príkazy **ftp**, **rexec**, **securetcpip** a **telnet** poskytujú nasledovné formy zabezpečenia systému a údajov:

ftp Príkaz **ftp** vytvára zabezpečené prostredie pre prenos súborov. Keď užívateľ vyvolá príkaz **ftp** na pripojenie k cudziemu hostiteľovi, zobrazí sa výzva, aby zadal prihlasovacie ID. Zobrazí sa predvolené prihlasovacie ID: aktuálne prihlasovacie ID užívateľa na lokálnom hostiteľovi. Užívateľovi sa zobrazí výzva na zadanie hesla pre vzdialeného hostiteľa.

Automatický prihlasovací proces prehľadá súbor $HOME/.netrc lokálneho užívateľa, aby identifikoval ID užívateľa a heslo, ktoré sa má použiť na cudzom hostiteľovi. Z dôvodu bezpečnosti musia byť oprávnenia na súbore $HOME/.netrc nastavené na hodnotu 600 (čítanie a zápis povolený len vlastníkovi). V opačnom prípade automatické prihlásenie zlyhá.

Poznámka: Keďže použitie súboru .netrc vyžaduje uskladnenie hesiel v nešifrovanom súbore, funkcia automatického prihlásenia príkazu **ftp** nie je k dispozícii, ak bol váš systém nakonfigurovaný príkazom **securetcpip**. Túto funkciu môžete opätovne povoliť, keď príkaz **ftp** odstránite zo state tcpip v súbore /etc/security/config.

Na používanie funkcie prenosu súborov príkaz **ftp** požaduje dve pripojenia TCP/IP - jedno pre protokol File Transfer Protocol (FTP) a druhé pre prenos údajov. Protokolové pripojenie je primárne a je bezpečné, pretože je vytvorené na spoľahlivých komunikačných portoch. Sekundárne pripojenie je potrebné pre samotný prenos údajov a rovnako lokálny a vzdialený hostiteľ overujú, či druhý koniec pripojenia je vytvorený s rovnakým hostiteľom ako primárne pripojenie. Ak primárne a sekundárne nie sú vytvorené s rovnakým hostiteľom, príkaz **ftp** najskôr zobrazí chybové hlásenie uvádzajúce, že dátové pripojenie nebolo autentifikované, a potom sa skončí. Toto overenie sekundárneho pripojenia bráni tretiemu hostiteľovi v zachytení údajov určených pre iného hostiteľa.

rexec Príkaz **rexec** poskytuje zabezpečené prostredie pre spúšťanie príkazov na cudzom hostiteľovi. Užívateľovi sa zobrazí výzva na zadanie prihlasovacieho ID aj hesla.

Funkcia automatického prihlasovania spôsobí, že príkaz **rexec** vyhľadá súbor $HOME/.netrc lokálneho užívateľa pre užívateľské ID a heslo na cudzom hostiteľovi. Z dôvodu bezpečnosti musia byť oprávnenia na súbore $HOME/.netrc nastavené na hodnotu 600 (čítanie a zápis povolený len vlastníkovi). V opačnom prípade automatické prihlásenie zlyhá.

Poznámka: Keďže použitie súboru .netrc vyžaduje uskladnenie hesiel v nešifrovanom súbore, funkcia automatického prihlásenia príkazu **rexec** nie je k dispozícii, ak váš systém funguje v zabezpečenom stave. Túto funkciu môžete opätovne povoliť, keď položku odstránite zo state tcpip v súbore /etc/security/config file.

securetcpip
Príkaz **securetcpip** povoľuje bezpečnostné funkcie protokolu TCP/IP. Prístup k príkazom, ktoré nie sú dôveryhodné, je zo systému odstránený pri zadaní takéhoto príkazu. Spustením príkazu **securetcpip** sa odstránia všetky nasledovné príkazy:

- **rlogin** a **rlogind**

- **rcp**, **rsh** a **rshd**
- **tftp** a **tftpd**
- **trpt**

Príkaz **securetcpip** sa používa na prepnutie systému zo štandardnej úrovne zabezpečenia na vyššiu úroveň zabezpečenia. Po prepnutí systému nie je nutné znovu zadávať príkaz **securetcpip**. Táto akcia je nutná iba v prípade, ak ste preinštalovali protokol TCP/IP.

telnet alebo tn

Príkaz **telnet** (TELNET) poskytuje zabezpečené prostredie pre prihlasovanie k cudziemu hostiteľovi. Užívateľovi sa zobrazí výzva na zadanie prihlasovacieho ID aj hesla. Vo vzťahu k terminálu užívateľa je správanie rovnaké ako v prípade terminálu priamo pripojeného k hostiteľovi. To znamená, že prístup k terminálu je riadený bitmi s oprávneniami. Ostatní užívatelia (skupina a iní) nemajú k terminálu prístup iba na čítanie, môžu mu však písať správy, ak im vlastník pridelí oprávnenia na zápis. Príkaz **telnet** tiež poskytuje prístup k dôveryhodnému prostrediu na vzdialenom systéme prostredníctvom funkcie SAK. Táto postupnosť klávesov sa líši od postupnosti, ktorá vyvoláva lokálnu dôveryhodnú cestu, a je ju možné definovať v rámci príkazu **telnet**.

Vzdialený prístup na spúšťanie príkazov:

Užívatelia na hostiteľoch, uvedených v súbore /etc/hosts.equiv, môžu na vašom systéme spúšťať určité príkazy bez zadania hesla.

Nasledujúca tabuľka poskytuje informácie o zobrazení, pridávaní a odstraňovaní vzdialených hostiteľov pomocou rozhrania nástroja SMIT alebo z rozhrania príkazového riadka.

Tabuľka 14. Úlohy vzdialeného prístupu na spúšťanie príkazov

Úloha	Rýchla cesta SMIT	Príkaz alebo súbor
Vypísanie vzdialených hostiteľov, ktorí majú prístup na spúšťanie príkazov	**smit lshostsequiv**	zobraziť súbor /etc/hosts.equiv
Pridanie vzdialeného hostiteľa pre prístup na spúšťanie príkazov	**smit mkhostsequiv**	upravte poznámku súboru /etc/hosts.equiv
Odstránenie vzdialeného hostiteľa z prístupu na spúšťanie príkazov	**smit rmhostsequiv**	upravte poznámku súboru /etc/hosts.equiv

Poznámka: Bližšie informácie o týchto súborových procedúrach nájdete v "hosts.equiv File Format for TCP/IP" v *Files Reference*.

Užívatelia obmedzeného programu na prenos súborov:

Užívatelia, uvedení v súbore /etc/ftpusers, sú chránení pred vzdialeným FTP prístupom. Predstavte si napríklad, že užívateľ A je prihlásený do vzdialeného systému a pozná heslo užívateľa B vo vašom systéme. Ak je užívateľ B uvedený v súbore /etc/ftpusers, užívateľ A nemôže využívať protokol FTP na prenos súborov z alebo do konta užívateľa B, hoci užívateľ A pozná heslo užívateľa B.

Nasledujúca tabuľka poskytuje informácie o zobrazení, pridávaní a odstraňovaní užívateľov s obmedzeniami pomocou rozhrania nástroja SMIT alebo z príkazového riadka.

Úlohy súvisiace s užívateľmi FTP

Úloha	Rýchla cesta SMIT	Príkaz alebo súbor
Vypísanie užívateľov FTP s obmedzeniami	**smit lsftpusers**	zobraziť súbor /etc/ftpusers
Pridanie užívateľa s obmedzeniami	**smit mkftpusers**	upravte poznámku súboru /etc/ftpusers
Odstránenie užívateľa s obmedzeniami	**smit rmftpusers**	upravte poznámku súboru /etc/ftpusers

Poznámka: Bližšie informácie o týchto súborových procedúrach nájdete v "ftpusers File Format for TCP/IP" v *Files Reference*.

Dôveryhodné procesy

Dôveryhodný program alebo dôveryhodný proces je skript prostredia, démon alebo program, ktorý vyhovuje konkrétnemu bezpečnostnému štandardu. Tieto bezpečnostné štandardy určuje a spravuje Ministerstvo obrany USA, ktoré taktiež certifikuje niektoré dôveryhodné programy.

Dôveryhodné programy sú dôveryhodné na rôznych úrovniach. Úrovne bezpečnosti zahŕňajú A1, B1, B2, B3, C1, C2 a D, pričom úroveň A1 predstavuje najvyššiu úroveň bezpečnosti. Každá úroveň bezpečnosti musí spĺňať určité požiadavky. Napríklad úroveň bezpečnosti C2 obsahuje nasledovné štandardy:

integrita programov
: Zabezpečuje, že procesy sa vykonávajú presne tak, ako je zamýšľané.

používanie modulov
: Zdrojový kód procesu je rozdelený do modulov, ktoré nemôžu byť priamo ovplyvnené inými modulmi a iné moduly k nim nemôžu priamo pristupovať.

princíp najnižšej úrovne oprávnení
: Určuje, že užívateľ pracuje vždy s pridelenou najnižšou úrovňou autorizovaných oprávnení. To znamená, že ak má užívateľ prístup len na prezeranie určitého súboru, potom neúmyselne nezíska prístup umožňujúci vykonať zmenu daného súboru.

obmedzenie opakovaného používania objektov
: Bráni užívateľovi, napríklad, v náhodnom nájdení časti pamäte, ktorá bola určená na prepis, ale ešte nebola vymazaná, a ktorá by mohla obsahovať citlivý materiál.

TCP/IP obsahuje niekoľkých dôveryhodných démonov a mnoho nedôveryhodných démonov.

Príklady dôveryhodných démonov:

- **ftpd**
- **rexecd**
- **telnetd**

Príklady nedôveryhodných démonov:

- **rshd**
- **rlogind**
- **tftpd**

Aby bo systém dôveryhodný, musí operovať s dôveryhodnou výpočtovou základňou. Pokiaľ berieme do úvahy jedného hostiteľa to znamená, že počítač musí byť zabezpečený. V prípade siete musia byť zabezpečené všetky súborové servery, brány a ostatní hostitelia.

Network Trusted Computing Base

Network Trusted Computing Base (NTCB) používa na zaistenie bezpečnosti siete hardvér a softvér. Táto časť definuje komponenty NTCB vo vzťahu k protokolu TCP/IP.

Funkcie starajúce sa o zabezpečenie hardvéru v sieti sú poskytované sieťovými adaptérmi používanými s protokolom TCP/IP. Tieto adaptéry riadia prichádzajúce údaje prijímaním len údajov smerovaných do lokálneho systému a vysielajú údaje prijateľné všetkými systémami.

Softvérový komponent súčasti NTCB pozostáva len z tých programov, ktoré sú považované za dôveryhodné. Programy a súvisiace súbory, ktoré sú súčasťou zabezpečeného systému, sú uvedené v nasledovných tabuľkách rozdelených podľa adresárov.

adresár /etc

Názov	Vlastník	Skupina	Režim	Oprávnenia
gated.conf	root	systém	0664	rw-rw-r—
gateways	root	systém	0664	rw-rw-r—
hosts	root	systém	0664	rw-rw-r—
hosts.equiv	root	systém	0664	rw-rw-r—
inetd.conf	root	systém	0644	rw-r—r—
named.conf	root	systém	0644	rw-r—r—
named.data	root	systém	0664	rw-rw-r—
networks	root	systém	0664	rw-rw-r—
protocols	root	systém	0644	rw-r—r—
rc.tcpip	root	systém	0774	rwxrwxr—
resolv.conf	root	systém	0644	rw-rw-r—
services	root	systém	0644	rw-r—r—
3270.keys	root	systém	0664	rw-rw-r—
3270keys.rt	root	systém	0664	rw-rw-r—

adresár /usr/bin

Názov	Vlastník	Skupina	Režim	Oprávnenia
host	root	systém	4555	r-sr-xr-x
hostid	bin	bin	0555	r-xr-xr-x
hostname	bin	bin	0555	r-xr-xr-x
finger	root	systém	0755	rwxr-xr-x
ftp	root	systém	4555	r-sr-xr-x
netstat	root	bin	4555	r-sr-xr-x
rexec	root	bin	4555	r-sr-xr-x
ruptime	root	systém	4555	r-sr-xr-x
rwho	root	systém	4555	r-sr-xr-x
talk	bin	bin	0555	r-xr-xr-x
telnet	root	systém	4555	r-sr-xr-x

adresár /usr/sbin

Názov	Vlastník	Skupina	Režim	Oprávnenia
arp	root	systém	4555	r-sr-xr-x
fingerd	root	systém	0554	r-xr-xr—
ftpd	root	systém	4554	r-sr-xr—
gated	root	systém	4554	r-sr-xr—
ifconfig	bin	bin	0555	r-xr-xr-x
inetd	root	systém	4554	r-sr-xr—
named	root	systém	4554	r-sr-x—

adresár /usr/sbin

Názov	Vlastník	Skupina	Režim	Oprávnenia
ping	root	systém	4555	r-sr-xr-x
rexecd	root	systém	4554	r-sr-xr—
route	root	systém	4554	r-sr-xr—
routed	root	systém	0554	r-xr-x—-
rwhod	root	systém	4554	r-sr-xr—
securetcpip	root	systém	0554	r-xr-xr—
setclock	root	systém	4555	r-sr-xr-x
syslogd	root	systém	0554	r-xr-xr—
talkd	root	systém	4554	r-sr-xr—
telnetd	root	systém	4554	r-sr-xr—

adresár /usr/ucb

Názov	Vlastník	Skupina	Režim	Oprávnenia
tn	root	systém	4555	r-sr-xr-x

adresár /var/spool/rwho

Názov	Vlastník	Skupina	Režim	Oprávnenia
rwho (adresár)	root	systém	0755	drwxr-xr-x

Bezpečnosť údajov a ochrana informácií

Funkcia zabezpečenia pre TCP/IP nešifruje sieťou prenášané užívateľské údaje.

Určite každé riziko v komunikácii, ktoré by mohlo mať za následok prezradenie hesiel a iných dôverných informácií a podľa tohto rizika použite príslušné protiopatrenia.

Používanie funkcie zabezpečenia TCP/IP v prostredí Ministerstva obrany USA (DOD) si vyžaduje dodržiavanie smerníc DOD 5200.5 a NCSD-11 pre bezpečnosť komunikácie.

Užívateľské riadenie prístupov na port TCP s Discretionary Access Control for Internet Ports (DACinet)

Funkcia Discretionary Access Control for Internet Ports (DACinet) umožňuje riadenie prístupu podľa užívateľov pre porty TCP pri komunikácii medzi hostiteľmi AIX.

Systém AIX môže používa ďalšiu hlavičku TCP na prenos informácií o užívateľoch a skupinách medzi systémami. Funkcia DACinet umožňuje administrátorovi na cieľovom systéme riadiť prístup na základe cieľového portu, pôvodného ID užívateľa a hostiteľa.

Okrem toho funkcia DACinet umožňuje administrátorovi obmedziť lokálne porty len pre používanie užívateľmi s oprávneniami typu root. Systémy UNIX podobne ako systémy AIX zaobchádzajú s portmi pod 1024 ako s privilegovanými portmi, ktoré môže otvárať len užívateľ s oprávneniami typu root. Systém AIX vám umožňuje určiť ďalšie porty nad 1024, ktoré môže otvoriť iba užívateľ root, vďaka čomu zabraňuje užívateľom v spúšťaní serverov na známych portoch.

V závislosti od nastavení sa systém bez funkcie DACinet môže, ale nemusí pripojiť k systému s funkciou DACinet. Prístup sa zakáže v úvodnej fáze behu funkcie DACinet. Len čo je funkcia DACinet povolená, nie je ju možné zakázať.

Príkaz **dacinet** akceptuje adresy, ktoré sú zadané ako názvy hostiteľov, desiatkové adresy hostiteľov s bodkou alebo sieťové adresy, za ktorými nasleduje údaj o dĺžke sieťového prefixu.

Nasledovný príklad uvádza jedného hostiteľa, ktorý je identifikovaný plne kvalifikovaným názvom *hostiteľ.doména.org*:

```
host.domain.org
```

Nasledovný príklad uvádza jedného hostiteľa, ktorý je identifikovaný adresou IP 10.0.0.1:

```
10.0.0.1
```

Nasledujúci príklad uvádza celú sieť, ktorá má prvých 24 bitov (dĺžka sieťovej predpony) s hodnotou 10.0.0.0:

```
10.0.0.0/24
```

Táto sieť zahŕňa všetky adresy IP od 10.0.0.1 do 10.0.0.254.

Riadenie prístupov pre služby založené na TCP:

DACinet používa spúšťací súbor /etc/rc.dacinet a konfiguračnými súbormi, ktoré používa, sú súbory /etc/security/priv, /etc/security/services a /etc/security/acl.

Porty uvedené v adresári /etc/security/services sa považujú za vyňaté z kontrol vykonávaných zoznamom ACL. Súbor má rovnaký formát ako /etc/services. Najjednoduchší spôsob jeho inicializácie je ten, že súbor skopírujete z adresára /etc do adresára /etc/security a potom odstránite všetky porty, pre ktoré by sa mali aplikovať zoznamy ACL. Zoznamy ACL sú uložené na dvoch miestach. Momentálne aktívne zoznamy ACL sú uložené v jadre a je ich možné prečítať spustením príkazu dacinet aclls. Zoznamy ACL, ktoré budú reaktivované pri ďalšom zavedení systému súborom /etc/rc.tcpip, sú uložené v adresári /etc/security/acl. Použije sa nasledovný formát:

```
hostiteľ služby/prefix-dĺžka [užívateľ|skupina]
```

Tam, kde možno zadať službu numericky alebo ako uvedenú v /etc/services, hostiteľa možno zadať ako názov hostiteľa alebo sieťovú adresu so zadaním masky podsiete a užívateľa a skupinu možno zadať s predponou u: alebo g:. Ak nie je zadaný žiadny užívateľ alebo skupina, zoznam ACL zohľadní len odosielajúceho hostiteľa. Ak pred službou použijete prefix -, služba sa zakáže explicitne. Zoznamy ACL sú vyhodnotené podľa prvej zhody. Môžete teda určiť prístup pre skupinu užívateľov, zároveň však explicitne zakázať tento prístup užívateľovi v skupine tým, že pravidlo pre tohto užívateľa umiestnite pred pravidlo skupiny.

Súbor /etc/services obsahuje dve položky s hodnotami čísel portov, ktoré nie sú podporované v systéme AIX. Systémový administrátor musí oba riadky z daného súboru odstrániť pred spustením príkazu **mkCCadmin**. Odstráňte nasledovné riadky zo súboru /etc/services:

```
sco_printer     70000/tcp     sco_spooler    # For System V print IPC
sco_s5_port     70001/tcp     lpNet_s5_port  # For future use
```

Príklady použitia DACinet:

Napríklad, keď pomocou DACinet obmedzíte prístup k vstupnému portu TCP/25 iba na užívateľa root s funkciou DACinet, k tomuto portu budú mať prístup iba užívatelia root z iných hostiteľov so systémom AIX, čím sa obmedzia možnosti štandardných užívateľov simulovať e-maily jednoduchým prístupom k portu TCP/25 prostredníctvom príkazu telnet na napadnutom počítači.

Nasledovný príklad uvádza spôsob konfigurácie protokolu X (X11) na prístup len pre užívateľov s oprávneniami typu root. Uistite sa, že položka X11 na ceste /etc/security/services je odstránená, takže pre túto službu sa použije zoznam prístupových práv.

Za predpokladu, že podsieť je pre všetky pripojené systémy 10.1.1.0/24, položky zoznamu prístupových práv pre obmedzenie prístupu užívateľa s oprávneniami typu root len na protokol X (TCP/6000) budú v súbore /etc/security/acl nasledovné:

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

Pri obmedzovaní služby Telnet na užívateľov v skupine friends bez ohľadu na to, z ktorého systému pochádzajú, použite po odstránení položky telnetu z /etc/security/services nasledujúcu položku ACL:

```
telnet    0.0.0.0/0   g:friends
```

Zrušte povolený prístup užívateľa fred k web serveru, ale povoľte prístup ostatným užívateľom:

```
-80    0.0.0.0/0 u:fred
80     0.0.0.0/0
```

Privilegované porty na spúšťanie lokálnych služieb:

Ak chcete zabrániť pravidelným užívateľom spúšťať servery na určitých portoch, môžete tieto porty vyhlásiť za privilegované.

Za normálnych okolností môže každý užívateľ otvoriť ľubovoľný port s číslom vyšším než 1024. Užívateľ môže napríklad umiestniť server na port 8080, ktorý sa pomerne často používa na spustenie web serverov proxy, alebo na port 1080, kde je bežne možné nájsť server SOCKS. Na pridanie privilegovaných portov do spusteného systému je možné použiť príkaz **dacinet setpriv**. Porty sa označia ako privilegované tak, aby sa pri spustení systému zobrazili v súbore /etc/security/priv.

Porty môžu byť v tomto súbore uvedené pod symbolickým názvom definovaným v súbore na ceste /etc/services alebo pod číslom portu. Nasledujúce položky znemožnia užívateľom, ktorí nemajú typ root, spúšťať servery SOCKS alebo servery Lotus Notes na svojich zvyčajných portoch:

```
1080
lotusnote
```

Poznámka: Táto funkcia nebráni užívateľovi user v spustení programov. Zabráni mu len v spúšťaní služieb na zvyčajných portoch, na ktorých sa tieto služby bežne spúšťajú.

Sieťové služby

Zobrazia sa informácie o identifikácii a zabezpečovaní sieťových služieb otvorenými komunikačnými portmi.

Použitie portov

Nasledujúca tabuľka popisuje použitie známych portov v operačnom systéme AIX.

Poznámka: Tento zoznam bol vytvorený preskúmaním viacerých systémov AIX s rôznymi konfiguráciami nainštalovaného softvéru.

Nasledujúci zoznam nemusí obsahovať použitie portov pre všetok softvér v operačnom systéme AIX:

Port/Protokol	Názov služby	Aliasy
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

Port/Protokol	Názov služby	Aliasy
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
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

Identifikácia sieťových služieb s otvorenými komunikačnými portmi

Aplikácie klient-server otvárajú komunikačné porty na serveri a umožňujú aplikáciám počúvať prichádzajúce požiadavky klientov.

Otvorené porty sú zraniteľné voči potenciálnym útokom na zabezpečenie, preto je nevyhnutné identifikovať, ktoré aplikácie majú otvorené porty a dané porty zatvoriť. Nasledovný postup je veľmi užitočný, pretože umožňuje porozumieť, ktoré systémy sa stávajú dostupné akejkoľvek osobe s prístupom na Internet.

Pri zisťovaní otvorených portov zvoľte tento postup:

1. Pomocou príkazu **netstat** identifikujte služby, ako je to uvedené nižšie:

   ```
   # netstat -af inet
   ```

Nižšie je uvedený príklad výstupu tohto príkazu. Posledný stĺpec výstupu príkazu **netstat** označuje stav každej služby. Služby, ktoré čakajú na prichádzajúce pripojenia sa nachádzajú v stave LISTEN.

Toto je príklad výstupu príkazu pri spustení príkazu **netstat**.

Aktívne internetové pripojenie (vrátane serverov)

Prot	Recv-Q	Send-Q	Lokálna adresa	Vzdialená adresa	(stav)
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
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. Otvorte súbor /etc/services a skontrolujte služby IANA (Internet Assigned Numbers Authority), čo umožňuje mapovať príslušné služby k číslam portov v rámci operačného systému.

   Nasleduje vzorový fragment súboru /etc/services:

```
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. Odstránením spustených služieb zatvorte nepotrebné porty.

Poznámka: Port 657 používa RMC (Resource Monitoring and Control) na komunikáciu medzi uzlami. Tento port nemôžete zablokovať alebo inak obmedziť.

Identifikácia soketov TCP a UDP

Na identifikáciu soketov TCP, ktoré sú v stave LISTEN a nečinných soketov UDP čakajúcich na príchod údajov použite príkaz **lsof**, ktorý je variantom príkazu **netstat -af**.

Ak chcete napríklad zobraziť sokety TCP v stave LISTEN a sokety UDP v stave IDLE, spustite príkaz **lsof** nasledovným spôsobom:

```
# lsof -i | egrep "COMMAND|LISTEN|UDP"
```

Výstup tohto príkazu je podobný nasledovnému:

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)

Po identifikácii ID procesu získate bližšie informácie o programe spustením príkazu:

```
" # ps -fp PID#"
```

Výstup obsahuje cestu k názvu príkazu, ktorú môžete použiť na získanie prístupu k hlavnej stránke programu.

Bezpečnosť internetového protokolu

Bezpečnosť IP povoľuje bezpečné komunikácie cez internet a v rámci podnikových sietí zabezpečením dátovej prevádzky na vrstve IP.

Prehľad IP Security

Bezpečnosť IP umožňuje jednotlivým užívateľom alebo organizátorom zabezpečiť prevádzku pre všetky aplikácie bez potreby vykonávať modifikácie aplikácií. Znamená to, že prenos akýchkoľvek údajov, ako sú napríklad e-maily alebo firemné údaje pre jednotlivé aplikácie, možno vykonávať bezpečne.

IP Security a operačný systém:

Operačný systém používa IP Security (IPsec), ktorá je otvorenou štandardnou bezpečnostnou technológiou vyvinutou Internet Engineering Task Force (IETF).

Technológia IPsec poskytuje kryptografickú ochranu všetkých údajov vo vrstve IP komunikačného zásobníka. V existujúcich aplikáciách nie je potrebné vykonať žiadne zmeny. Technológia IPsec predstavuje štandardizovaný rámec zabezpečenia sietí zvolený združením IETF pre prostredia s protokolom IPv4 aj IPv6.

Technológia IPsec používa na ochranu prenosu údajov nasledovné kryptografické techniky:

Autentifikácia
: Proces na overenie totožnosti hostiteľa alebo koncového bodu

Kontrola integrity
: Proces, v ktorom sa overuje, že počas prenosu v sieti nedošlo k modifikácii údajov

Šifrovanie
: Proces na zaistenie ochrany osobných údajov "skrytím" týchto údajov a súkromných adries IP počas prenosu v sieti

Autentifikačné algoritmy overujú totožnosť odosielateľa a integritu údajov prostredníctvom kryptografickej hašovacej funkcie, ktorá spracovaním paketu údajov (vrátane fixovaných polí hlavičky IP) s použitím tajného kľúča vytvorí jedinečný súhrn. Na strane príjemcu sa na spracovanie údajov použije tá istá funkcia a kľúč. Ak boli údaje pozmenené alebo kľúč odosielateľa nie je platný, datagram sa vymaže.

Šifrovanie používa kryptografický algoritmus na úpravu údajov podľa určitého algoritmu a kľúč na vytvorenie šifrovaných údajov (označujú sa ako *cyphertext*. Šifrovanie znemožňuje čitateľnosť údajov počas prenosu. Po prijatí sa údaje obnovia s použitím toho istého algoritmu a kľúča (algoritmy pre symetrické šifrovanie). Šifrovanie musí byť úzko späté s autentifikáciou, aby sa overila integrita zašifrovaných údajov.

Tieto základné služby sú implementované v technológii IPsec prostredníctvom protokolov Encapsulating Security Payload (ESP) a Authentication Header (AH). Protokol ESP zabezpečuje dôvernosť zašifrovaním pôvodného paketu IP, vytvorením hlavičky ESP a vložením údajov vo formáte cyphertext do prenosu ESP.

Protokol AH sa môže použiť na autentifikáciu a kontrolu integrity aj samostatne, ak sa nekladie dôraz na dôvernosť. Ak sa použije protokol AH, na statické polia hlavičky IP a na údaje sa použije hašovací algoritmus, ktorý vytvorí súhrn s použitím kľúča. Príjemca použije svoj kľúč na vyčíslenie a porovnanie súhrnu, aby sa uistil, že paket je nezmenený a totožnosť odosielateľa je autentifikovaná.

Funkcie IP Security:

Nasledujú funkcie bezpečnosti IP.

Nasledujúce funkcie sú k dispozícii s Internet Key Exchange pre operačný systém AIX:

- Podporuje 128-bitové, 192-bitové a 256-bitové algoritmy AES.
- Hardvérové urýchľovanie s adaptérom 10/100 Mbps Ethernet PCI Adapter II.
- Podpora pre AH prostredníctvom RFC 2402 a podpora pre ESP prostredníctvom RFC 2406.
- Manuálne tunely možno nakonfigurovať na poskytovanie interoperability s ostatnými systémami, ktoré nepodporujú metódu automatickej obnovy kľúčov IKE a na použitie tunelov IP verzie 6.
- Režim tunela a režim prenosu enkapsulácie pre tunely hostiteľa alebo brány.

- Algoritmy autentifikácie HMAC (Hashed Message Authentication Code) MD5 (Message Digest 5) a HMAC SHA (Secure Hash Algorithm).
- Šifrovacie algoritmy zahŕňajú 56-bitové šifrovanie DES (Data Encryption Standard) CBC (Cipher Block Chaining) s 64-bitovým šifrovaním IV (Initial Vector), Triple DES, DES CBC 4 (32-bitové IV) a AES CBC.
- Podpora duálneho zásobníka IP (IP verzia 4 a 6).
- Možnosť enkapsulácie a filtrácie prenosu s použitím protokolov IPv4 a IPv6. Keďže sú IP zásobníky samostatné, funkcia IP Security pre každý zásobník sa môže posudzovať nezávisle.
- Filtrovanie zabezpečených aj nezabezpečených prenosov podľa rôznych charakteristík IP, ako sú zdrojová a cieľová IP adresa, rozhranie, protokol, čísla portov a iné.
- Automatické vytvorenie a vymazanie pravidiel filtra u väčšiny typov tunelov.
- Používanie názvov hostiteľov pre cieľovú adresu pri definovaní tunelov a pravidiel filtrovania. Názvy hostiteľov sa skonvertujú na IP adresy automaticky (ak je k dispozícii server DNS).
- Protokolovanie udalostí IP Security do súboru **syslog**.
- Použitie systémových stôp a štatistiky na zisťovanie problémov.
- Užívateľom definovaná predvolená akcia umožňuje užívateľom určiť, či sú povolené prenosy nevyhovujúce definovaným tunelom.

Nasledujúce doplnkové funkcie sú dostupné s IKE (Internet Key Exchange) pre verziu AIX 6.1 TL 05 alebo novšie:
- Podpora pre IPSec prostredníctvom RFC 4301, podpora pre AH prostredníctvom RFC 4302 a podpora pre ESP prostredníctvom RFC 4303
- Autentifikačné algoritmy CMAC (Cipher-based Message Authentication Code) AES XCBC
- Šifrovacie algoritmy zahŕňajú 128-bitový algoritmu AES, 192-bitový a 256-bitový algoritmus GCM (16-bitový IV), AES-128-GMAC, AES-192-GMAC a AES-256-GMAC
- Podpora rozsahu portov pre pravidlá filtrov
- Čísla rozšírenej postupnosti

Funkcie Internet Key Exchange:

Nasledujú funkcie, ktoré sú dostupné pomocou IKE (Internet Key Exchange) pre AIX.

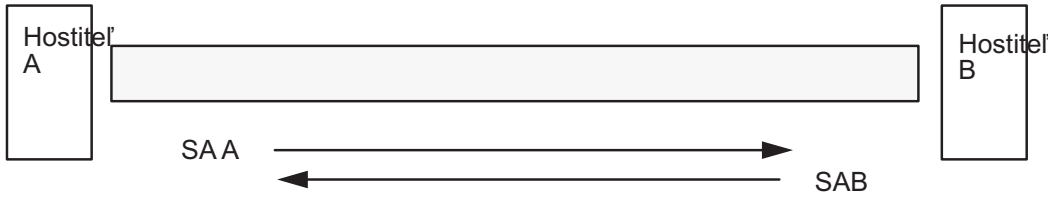
Nasledujúce doplnkové funkcie sú dostupné s IKE (Internet Key Exchange) pre verziu AIX 6.1 alebo novšie:
- Podpora AH pre 256-bitový haš HMAC SHA2 (verzia TL 04 alebo novšie).
- Podpora šifrovania ESP 128-bitový, 192-bitový, 256-bitový GCM AES s (16-bitový IV), 128-bitové, 192-bitové, 256-bitové algoritmy GMAC AES; podpora autentifikácie ESP s HMAC MD5 a HMAC SHA1 (verzia TL 04 alebo novšie).
- IKEv1 (RFC2409) a IKEv2 (RFC4306) sú podporované (verzia TL 02 alebo novšie). IKEv1 podporuje démon **isakmpd** a IKEv2 podporuje démon **ikev2d** (verzia TL 02 alebo novšie). Tunely IKEv1 a IKEv2 môžu existovať súčasne.
- Podpora algoritmov integrity CMAC_AES_XCBC a HMAC_SHA2_256 (verzia TL 04 alebo novšie).
- Podpora algoritmu PRF PRF_SHA2_256 (verzia TL 04 alebo novšie).
- Podpora skupín Diffie Hellman 14, 19 a 24 (verzia TL 04 alebo novšie).

Priradenia bezpečnosti:

Zabezpečená komunikácia vychádza z koncepcie tzv. *priradení zabezpečenia*. Priradenia bezpečnosti vzťahujú určitú sadu bezpečnostných parametrov k určitému typu prevádzky.

Ak sú údaje chránené technológiou IP Security, existuje osobitné priradenie zabezpečenia pre každý smer a pre každý typ hlavičky (AH alebo ESP). Informácie obsiahnuté v priradení zabezpečenia zahŕňajú adresy IP komunikujúcich

strán, jedinečný identifikátor SPI (Security Parameters Index), algoritmy vybraté pre autentifikáciu alebo šifrovanie, autentifikačné a šifrovacie kľúče a životnosti kľúčov. Na nasledujúcej ilustrácii sú znázornené priradenia zabezpečenia medzi hostiteľom A a hostiteľom B.



SA = bezpečnostná asociácia, skladá sa z:

Cieľová adresa
SPI
Kľúč
Šifrovací algoritmus a formát
Autentifikačný algoritmus
Životnosť kľúča

Obrázok 6. Vytvorenie zabezpečeného tunelového prepojenia medzi hostiteľom A a hostiteľom B

Tento obrázok zobrazuje virtuálny tunel medzi hostiteľom A a hostiteľom B. Priradenie bezpečnosti A je šípkou smerujúcou z hostiteľa A na hostiteľa B. Priradenie bezpečnosti B je šípkou smerujúcou z hostiteľa B na hostiteľa A. Priradenie bezpečnosti pozostáva z cieľovej adresy, SPI, kľúča, šifrovacieho algoritmu a formátu, algoritmu autentifikácie a životnosti kľúča.

Cieľom správy kľúčov je vyjednať a vyčísliť priradenia zabezpečenia, ktoré ochránia prenos IP.

Tunely a správa kľúčov:

Tunel použite na vyjednávanie a riadenie priradení bezpečnosti potrebných na nastavenie bezpečnej komunikácie medzi dvomi hostiteľmi.

Podporované sú tieto typy tunelov, z ktorých každý používa inú techniku správy kľúčov:

- Tunelové prepojenia IKE (dynamicky sa meniace kľúče, štandard IETF)
- Manuálne tunelové prepojenia (statické a trvalé kľúče, štandard IETF)

Podpora tunela Internet Key Exchange:

Tunely IKE sú založené na Internet Security Association and Key Management Protocol (ISAKMP)/štandardoch Oakley vyvinutých IETF. S týmto protokolom sa vyjednávajú a obnovujú bezpečnostné parametre a kľúče sa vymieňajú bezpečným spôsobom.

Podporované sú nasledujúce typy autentifikácie:

- Dopredu zdieľaný kľúč.
- Podpisy digitálnych certifikátov X.509v3.
- IKEv2 vo verzii AIX 6.1 TL 04 alebo novšej podporuje podpisy digitálnych certifikátov ECDSA-256 ako súčasť metódy autentifikácie X509v3, ktorá je založená na digitálnych certifikátoch.

Vyjednávanie pozostáva z dvoch fáz. Fáza 1 autentifikuje komunikujúce strany a špecifikuje algoritmy, ktoré sa majú použiť na bezpečnú komunikáciu vo fáze 2. Počas fázy 2 sa vyjednávajú parametre IP Security, ktoré sa majú použiť počas prenosu údajov, a vytvoria a vymenia sa bezpečnostné asociácie a kľúče.

Autentifikačné algoritmy, ktoré sa môžu použiť s protokolmi AH a ESP pre podporu tunelových prepojení IKE, sú uvedené v nasledovnej tabuľke.

Tabuľka 15. Algoritmy autentifikácie pre podporu tunela IKE

Algoritmus	AH IPv4 & 6	ESP IPv4 & 6
HMAC MD5	X	X
HMAC SHA1	X	X
DES CBC 8		X
Triple 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

Podpora manuálneho tunelového prepojenia:

Manuálne tunelové prepojenia zabezpečujú spätnú kompatibilitu a spoluprácu s počítačmi, ktoré nepodporujú protokoly pre správu kľúčov IKE. Nevýhodou manuálnych tunelových prepojení sú statické hodnoty kľúčov. Šifrovacie a autentifikačné kľúče sú rovnaké pre životnosť tunela a musia sa aktualizovať manuálne.

Autentifikačné algoritmy, ktoré sa môžu použiť s protokolmi AH a ESP na podporu manuálnych tunelových prepojení, sú uvedené v nasledovnej tabuľke.

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
Triple DES CBC			X	X
DES CBC 8			X	X
DES CBC 4			X	X

Preferovanou metódou správy kľúčov je výmena IKE, pretože tunelové prepojenia IKE ponúkajú účinnejšie zabezpečenie.

Natívna filtrovacia schopnosť:

Filtrovanie je základná funkcia, pri ktorej sa prichádzajúce a odchádzajúce pakety môžu prijať alebo odmietnuť na základe rôznych charakteristík. Umožňuje to užívateľovi alebo správcovi systému nakonfigurovať hostiteľa na riadenie prevádzky medzi týmto a ostatnými hostiteľmi.

Filtrovanie sa vykonáva na základe rôznych vlastností paketov, ako sú napríklad zdrojové a cieľové adresy, verzia protokolu IP (4 alebo 6), masky podsiete, protokoly, porty, charakteristiky smerovania, fragmentácia, rozhranie a definícia tunelového prepojenia.

Pravidlá, ktoré sa označujú ako *pravidlá filtra*, slúžia na priradenie určitých typov prenosu k určitému tunelovému prepojeniu. V základnej konfigurácii manuálnych tunelových prepojení sa pri definovaní tunelového prepojenia hostiteľ-hostiteľ automaticky generujú pravidlá filtra, ktoré smerujú celý prenos z hostiteľa cez zabezpečené tunelové

prepojenie. Ak sa požadujú špecifickejšie typy prenosu (napríklad z podsiete do podsiete), pravidlá filtra je možné upraviť alebo nahradiť a umožniť tak presnejšie riadenie prenosu používajúceho príslušné tunelové prepojenie.

V prípade tunelových prepojení IKE sa pravidlá filtra tiež generujú automaticky a vložia sa do tabuľky filtrov po aktivácii tunelového prepojenia.

Pri modifikácii alebo odstránení tunelového prepojenia sa automaticky odstránia aj pravidlá filtra pre toto prepojenie, vďaka čomu sa zjednoduší konfigurácia IP Security a redukuje sa možnosť vzniku chyby užívateľa. Definície tunelových prepojení je možné preniesť a zdieľať medzi počítačmi a bránami firewall prostredníctvom pomocných programov pre import a export, čo je užitočné najmä pri správe veľkého počtu počítačov.

Pravidlá filtra priraďujú určité typy prenosov k tunelovému prepojeniu, ale filtrované údaje nemusia nevyhnutne prechádzať tunelovým prepojením. Tento aspekt pravidiel filtra umožňuje operačnému systému poskytovať základné funkcie brány firewall užívateľom, ktorí chcú obmedziť prenos z počítača a do počítača na intranete alebo v sieti nechránenej skutočnou bránou firewall. V tomto prípade pravidlá filtra predstavujú druhú ochrannú bariéru určitej skupiny počítačov.

Pravidlá filtra sa po vygenerovaní uložia do tabuľky a načítajú do jadra. Keď sú pakety pripravené na odoslanie alebo prijatie zo siete, na základe kontroly podľa pravidiel filtra v zozname sa určuje, či sa paket má povoliť, zakázať alebo odoslať cez tunelové prepojenie. Kritériá pravidiel sa porovnávajú s charakteristikami paketu, kým sa nenájde zhoda, alebo až do vyhľadania predvoleného pravidla.

Táto funkcia IP Security implementuje aj filtrovanie nezabezpečených paketov na základe približných kritérií definovaných užívateľom, ktoré umožňujú riadiť prenos IP medzi sieťami a počítačmi, ktoré nepožadujú autentifikačné a šifrovacie funkcie technológie IP Security.

Podpora digitálnych certifikátov:

IP Security podporuje použitie digitálnych certifikátov X.509 verzie 3.

Nástroj Key Manager spravuje žiadosti o certifikáty, vykonáva údržbu databázy kľúčov a vykonáva aj ďalšie administratívne činnosti.

Digitálne certifikáty sú popísané v časti Konfigurácia digitálnych certifikátov. Nástroj Key Manager a jeho funkcie sú opísané v časti Using the IBM Key Manager Tool

Virtuálne súkromné siete (VPN) a IP Security:

Virtuálna súkromná sieť (VPN) bezpečne rozširuje súkromný intranet vo verejnej sieti akou je napríklad internet.

Informácie sa v týchto sieťach prenášajú oboma smermi v podstate cez súkromné internetové tunelové prepojenie na vzdialených užívateľov, pobočky, obchodných partnerov a dodávateľov. Spoločnosti sa môžu rozhodnúť pre prístup na Internet od poskytovateľov internetových služieb prostredníctvom priamych liniek alebo miestnych telefónnych čísel a vylúčiť tak nákladnejšie prenajaté linky, medzimestské hovory a bezplatné telefónne čísla. Riešenie s virtuálnou súkromnou sieťou môže využiť štandard IPsec, pretože ide o štandardizovaný rámec IETF pre sieťové zabezpečenie v prostredí s protokolom IPv4 a IPv6, ktorý nevyžaduje žiadne zmeny v existujúcich aplikáciách.

Odporúčaným zdrojom pre naplánovanie implementácie siete VPN v operačnom systéme AIX je kapitola 9 príručky *A Comprehensive Guide to Virtual Private Networks, Volume III: Cross-Platform Key and Policy Management*, ISBN SG24-5309-00. Táto príručka je tiež dostupná na internete na adrese http://www.redbooks.ibm.com/redbooks/SG245309.html.

Inštalácia funkcie IP Security

Funkciu IP Security možno v systéme AIX nainštalovať a zaviesť oddelene.

Sady súborov musia byť nainštalované nasledujúcim spôsobom:

- bos.net.ipsec.rte (prostredie runtime pre prostredie a príkazy zabezpečenia IP jadra)
- bos.msg.*JAZYK*.net.ipsec (kde *JAZYK* predstavuje určený jazyk, napríklad en_US)
- bos.net.ipsec.keymgt
- clic.rte (CryptoLite pre C, sady súborov pre DES, trojité DES a šifrovanie AES)

Pre podporu digitálnych podpisov IKE musíte nainštalovať aj sadu súborov gskit.rte alebo gskkm.rte z balíka Expansion Pack.

Po nainštalovaní sa môže zabezpečenie IP zaviesť samostatne pre protokoly IPv4 a IPv6, či už vykonaním odporúčaného postupu uvedeného v časti "Zavedenie modulov IP Security", alebo pomocou príkazu **mkdev**.

Zavedenie modulov IP Security:

Bezpečnostné moduly IP môžete zaviesť po spustení súčasti IP Security pomocou nástroja SMIT. Nástroj SMIT tiež zabezpečí zavedenie rozšírení jadra a démonov IKE v správnom poradí.

Poznámka: Zavedenie IP Security zapína funkciu filtrovania. Pred zavedením tejto funkcie je potrebné skontrolovať, či sú vytvorené správne pravidlá filtrovania. V opačnom prípade môže nastať zablokovanie komunikácie s vonkajším prostredím.

V prípade úspešného zavedenia príkaz **lsdev** zobrazí zariadenia IP Security ako Available.

```
lsdev -C -c ipsec

   ipsec_v4 Available IP Version 4 Security Extension
   ipsec_v6 Available IP Version 6 Security Extension
```

Po načítaní rozšírenia jadra funkcie IP Security sú tunely a filtre pripravené na konfiguráciu.

Plánovanie konfigurácie IP Security

Ak chcete nakonfigurovať IP Security, naplánujte si najprv konfiguráciu tunelových prepojení a filtrov.

Pri definovaní používania jediného tunelového prepojenia pre celý prenos sa pravidlá pre filtrovanie môžu generovať automaticky. Ak sa požaduje zložitejšie filtrovanie, pravidlá pre filtrovanie je možné nakonfigurovať samostatne.

IP Security môžete nakonfigurovať prostredníctvom modulu plug-in Virtual Private Network alebo nástroja SMIT (System Management Interface Tool). Ak použijete nástroj SMIT, k dispozícii sú tieto rýchle cesty:

smit ips4_basic
: Základná konfigurácia pre protokol IP Verzia 4

smit ips6_basic
: Základná konfigurácia pre protokol IP Verzia 6

Pred nakonfigurovaním protokolov IP Security sa musíte rozhodnúť, ktorý spôsob chcete používať, napríklad či chcete používať radšej tunelové prepojenia alebo filtre (alebo oboje), ktorý typ tunelového prepojenia považujete za najvhodnejší, atď. Skôr, než sa rozhodnete, mali by ste sa oboznámiť s informáciami uvedenými v nasledovných sekciách:

Akcelerácia hardvéru:

10/100 Mbps Ethernet PCI Adapter II (kód funkcie 4962) ponúka štandardnú bezpečnosť IP a je navrhnutý na offload bezpečnostnej funkcie IP z operačného systému AIX.

Ak sa tento adaptér nachádza v systéme AIX, zásobník protokolov IP Security využíva tieto schopnosti adaptéra:
- Šifrovanie a dešifrovanie s použitím algoritmov DES alebo Triple DES

- Autentifikácia s použitím algoritmov MD5 alebo SHA-1
- Ukladanie bezpečnostných informácií

Namiesto softvérových algoritmov sa používajú funkcie adaptéra. 10/100 Mbps Ethernet PCI Adapter II je k dispozícii pre manuálne a IKE tunely.

Funkcia akcelerácie hardvéru bezpečnosti IP je dostupná v 5.1.0.25 alebo novšej úrovni sád súborov bos.net.ipsec.rte a devices.pci.1410ff01.rte.

Počet priradení zabezpečenia, o ktoré je možné znížiť záťaž sieťového adaptéra na strane príjemcu (prichádzajúci prenos), je limitovaný. Na strane odosielateľa (odchádzajúci prenos) sa záťaž znižuje presunom všetkých paketov používajúcich podporovanú konfiguráciu na adaptér. Niektoré konfigurácie tunelových prepojení neumožňujú presunutie záťaže na adaptér.

10/100 Mbps Ethernet PCI Adapter II podporuje tieto funkcie:
- Šifrovanie DES, 3DES alebo NULL s použitím ESP
- Autentifikáciu HMAC-MD5 alebo HMAC-SHA-1 s použitím ESP alebo AH, ale nie oboje (Ak sa používa ESP aj AH, prednostne sa musí použiť ESP - platí to v prípade tunelových prepojení IKE, pri manuálnych tunelových prepojeniach môže užívateľ zvoliť poradie.)
- Režim prenosu a tunelového prepojenia
- Zníženie záťaže o pakety IPV4

Poznámka: Adaptér 10/100 Mbps Ethernet PCI Adapter II nemôže spracovať pakety s voľbami IP.

Ak chcete aktivovať adaptér 10/100 Mbps Ethernet PCI Adapter II na používanie protokolov IP Security, bude zrejme nutné odpojiť sieťové rozhranie a aktivovať funkciu IPsec Offload.

Pri odpájaní sieťového rozhrania vykonajte pomocou rozhrania SMIT tieto kroky:

Ak chcete aktivovať funkciu IPsec Offload, použite v rozhraní nástroja SMIT tento postup:
1. Prihláste sa ako užívateľ typu **root**.
2. Do príkazového riadka zadajte smitty eadap a stlačte kláves Enter.
3. Vyberte voľbu **Change / Show Characteristics of an Ethernet Adapter** a stlačte kláves Enter.
4. Vyberte adaptér 10/100 Mbps Ethernet PCI Adapter II a stlačte kláves Enter.
5. Zmeňte hodnotu pre pole IPsec Offload na yes a stlačte kláves Enter.

Ak chcete odpojiť sieťové rozhranie z príkazového riadka, napíšte:
```
# ifconfig enX detach
```

Ak chcete povoliť atribút IPsec offload do príkazového riadka napíšte:
```
# chdev -l entX -a ipsec_offload=yes
```

Ak chcete skontrolovať, či bol atribút IPsec offload zapnutý, do príkazového riadka napíšte:
```
# lsattr -El entX detach
```

Ak chcete vypnúť atribút IPsec offload do príkazového riadka napíšte:
```
# chdev -l entX -a ipsec_offload=no
```

Ak sa chcete uistiť, že konfigurácia tunelového prepojenia využíva funkciu IPsec Offload, použite príkaz **enstat**. Ak je atribút IPsec Offload aktivovaný, príkaz **enstat** zobrazuje úplné štatistické údaje o vysielaní a prijímaní paketov IPsec. Napríklad v prípade rozhrania siete Ethernet **ent1** zadajte:
```
# entstat -d ent1
```

Výstup bude podobný tomuto:

```
.
.
.
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
```

Ladiaci parameter siete:

V závislosti od počtu tunelov vo vašej konfigurácii môžete zvýšiť maximálnu veľkosť vyrovnávacej pamäte pre soket.

Ak je vo vašom prostredí spustený veľký počet tunelov a nezmeníte predvolenú hodnotu ladiaceho parametra **sb_max**, démonový proces IKE a démonový proces Tunnel Manager môžu prestať odpovedať v dôsledku vysokého zaťaženia siete.

Hodnotu ladiaceho parametra **sb_max** odporúčame nastaviť takto:

- 10 MB pre 500 tunelov
- 20 MB pre 1000 tunelov

Súvisiace informácie:

Ladiaci parameter sb_max

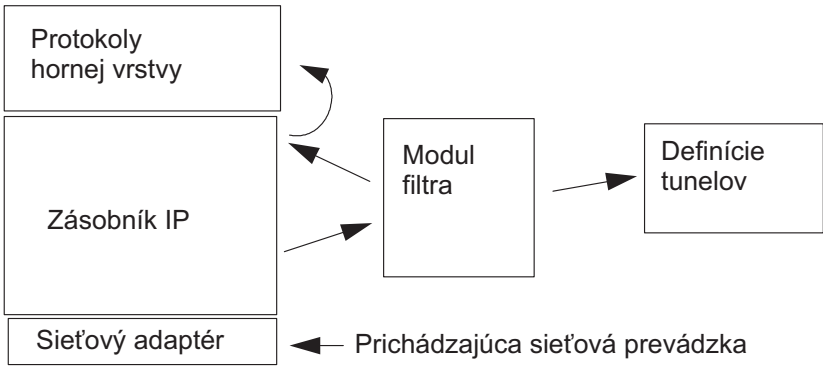
Tunely verzus filtre:

Dve rozdielne súčasti zabezpečenia IP sú *tunely* a *filtre*. Tunelové prepojenia vyžadujú filtre, ale filtre nevyžadujú tunelové prepojenia.

Filtrovanie je funkcia, pri ktorej sa prichádzajúce a odchádzajúce pakety môžu prijať alebo odmietnuť na základe rôznych charakteristík, ktoré sa označujú ako *pravidlá*. Táto funkcia umožňuje systémovému administrátorovi nakonfigurovať hostiteľa na riadenie prenosu medzi týmto hostiteľom a ostatnými hostiteľmi. Filtrovanie sa vykonáva na základe rôznych vlastností paketov, ako sú napríklad zdrojové a cieľové adresy, verzia protokolu IP (4 alebo 6), masky podsiete, protokoly, porty, charakteristiky smerovania, fragmentácia, rozhranie a definícia tunelového prepojenia. Toto filtrovanie prebieha na vrstve protokolu IP, takže žiadne zmeny aplikácií sa nevyžadujú.

Tunelové prepojenia definujú priradenie zabezpečenia medzi dvoma hostiteľmi. Tieto priradenia zabezpečenia zahŕňajú špecifické parametre zabezpečenia, ktoré sa zdieľajú medzi dvoma koncovými bodmi tunelového prepojenia.

Na nasledovnej ilustrácii je znázornená cesta paketu zo sieťového adaptéra do zásobníka protokolu IP. V zásobníku sa vyvolá modul filtra, ktorý určí, či ide o povolený alebo zakázaný paket. Ak je zadané ID tunelového prepojenia, paket sa overuje voči existujúcim definíciám tunelových prepojení. Ak je dekapsulácia z tunelového prepojenia úspešná, paket sa prenesie do protokolu vyššej úrovne. Pri odchádzajúcich paketoch sa používa opačná postupnosť krokov. Tunelové prepojenie používa pravidlo filtrovania na priradenie paketu k príslušnému prepojeniu, ale filtrovanie sa môže vyskytnúť aj bez presunutia paketu do tunelového prepojenia.



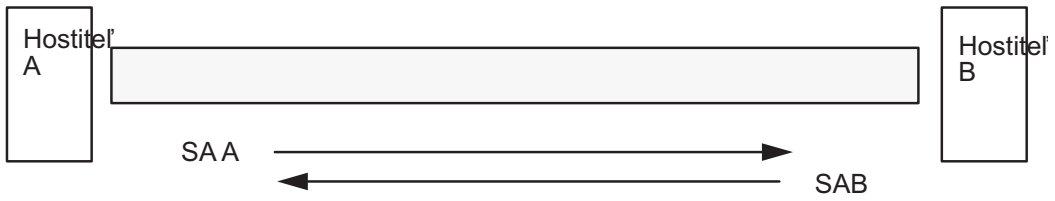
Obrázok 7. Smerovanie sieťových paketov

Ilustrácia ukazuje trasu sieťového paketu. Ako prichádzajú zo siete, pakety vstupujú do sieťového adaptéra. Odtiaľ prechádza do zásobníka protokolu IP, z ktorého sa odosiela do modulu filtra. Z modulu filtra sa paket odošle na overenie definíciou tunelového prepojenia alebo sa vráti do zásobníka protokolu IP, kde prejde do protokolov vyššej vrstvy.

Tunely a priradenia zabezpečenia:

Tunelové prepojenia sa používajú v prípade, že údaje musia byť autentifikované alebo autentifikované a zašifrované. Tunelové prepojenia sú definované zadaním priradenia zabezpečenia medzi dvoma hostiteľmi. Priradenie zabezpečenia definuje parametre pre algoritmy šifrovania a autentifikácie a charakteristiky tunelu.

Na nasledujúcej ilustrácii je znázornené tunelové prepojenie medzi hostiteľom A a hostiteľom B.



SA = bezpečnostná asociácia, skladá sa z:

Cieľová adresa
SPI
Kľúč
Šifrovací algoritmus a formát
Autentifikačný algoritmus
Životnosť kľúča

Obrázok 8. Vytvorenie zabezpečeného tunelového prepojenia medzi hostiteľom A a hostiteľom B

Ilustrácia ukazuje virtuálny tunel medzi hostiteľom A a hostiteľom B. Priradenie zabezpečenia A je šípka smerujúca od hostiteľa A k hostiteľovi B. Priradenie zabezpečenia B je šípka smerujúca od hostiteľa B k hostiteľovi A. Priradenie zabezpečenia pozostáva z cieľovej adresy, SPI, kľúča, šifrovacieho algoritmu a formátu, algoritmu autentifikácie a životnosti kľúča.

Index parametrov zabezpečenia SPI (Security Parameter Index) a cieľová adresa určujú jedinečné priradenie zabezpečenia. Tieto parametre sú nevyhnutné na jedinečnú špecifikáciu tunelového prepojenia. Ďalšie parametre, ako algoritmus šifrovania, algoritmus autentifikácie, kľúče a životnosť, môžu byť zadané alebo použité s predvolenými hodnotami.

Veci, ktoré treba brať do úvahy pri tuneloch:

Skôr ako sa rozhodnete, ktorý typ tunelu použijete pre zabezpečenie IP, mali by ste zvážiť niekoľko vecí.

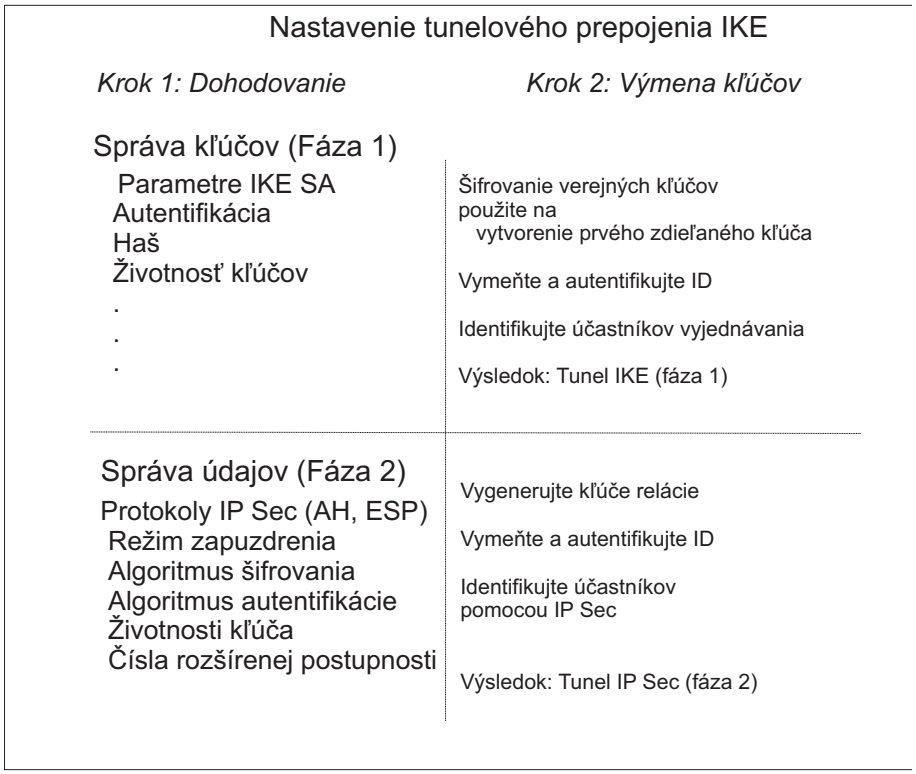
Tunely IKE sa líšia od manuálnych tunelov, pretože konfigurácia bezpečnostných politík je samostatný proces, oddelený od definovania koncových bodov tunelu.

V prípade výmeny IKE proces vyjednávania pozostáva z dvoch krokov. Jednotlivé kroky procesu vyjednávania sa označujú ako *fázy* a každá fáza sa môže riadiť osobitnou politikou bezpečnosti.

Pri spustení vyjednávania výmeny kľúčov sa musí vytvoriť zabezpečený kanál pre vyjednávania. Tento proces sa označuje ako fáza *správy kľúčov* alebo *fáza 1*. V tejto fáze každá zo strán použije vopred zdieľané kľúče alebo digitálne certifikáty na autentifikáciu druhej strany a na odoslanie informácií o ID. V tejto fáze sa vytvorí priradenie zabezpečenia, počas ktorého dve strany určia spôsob bezpečnej komunikácie a spôsoby ochrany komunikácie počas druhej fázy. Výsledkom akcií vykonaných počas tejto fázy je tunelové prepojenie *IKE* alebo tunelové prepojenie *fázy 1*.

Druhá fáza sa označuje ako fáza *správy údajov* alebo *fáza 2* a využíva tunelové prepojenie IKE na vytvorenie priradení zabezpečenia pre AH a ESP, ktoré v skutočnosti ochraňujú prenos. V druhej fáze sa určujú aj údaje, ktoré budú používať tunelové prepojenie protokolov IP Security. Určené môžu byť napríklad:

- Maska podsiete
- Rozsah adries
- Kombinácia protokolu a čísla portu



Obrázok 9. Nastavenie tunelového prepojenia IKE

Na tejto ilustrácii je znázornený dvojfázový proces nastavenia tunelového prepojenia IKE (pozostáva z dvoch krokov).

Poznámka: IKEv2 má tiež dve fázy. Prvá fáza sa označuje fáza *IKE SA* alebo *phase 1*. Druhá fáza sa označuje ako fáza *CHILD SA* alebo *phase 2* . Na rozdiel od spôsobu, akým sa vytvárajú tunelové prepojenia v IKEv1, pri vytvorení

tunelového prepojenia fázy 1 v IKEv2 sa automaticky aktivuje tunelové prepojenie fázy 2. Konfigurácia tunelových prepojení IKEv2 je podobná konfigurácii tunelových prepojení IKEv1.

Koncové body tunelového prepojenia pre správu kľúčov (IKE) sú často totožné s koncovými bodmi tunelového prepojenia pre správu údajov (IP Security). Koncové body tunelového prepojenia IKE sú ID počítačov zabezpečujúcich vyjednávanie. Koncové body tunelového prepojenia IP Security popisujú typ prenosu, pri ktorom sa bude používať tunelové prepojenie IP Security. V prípade jednoduchých tunelových prepojení typu hostiteľ-hostiteľ, pri ktorých je prenos medzi dvoma tunelovými prepojeniami chránený jedným tunelovým prepojením, sú koncové body tunelových prepojení fázy 1 a fázy 2 totožné. Ak sú vyjednávajúcimi stranami dve brány, koncovými bodmi tunelového prepojenia IKE sú tieto dve brány a koncovými bodmi tunelového prepojenia IP Security sú počítače, podsiete alebo určitý rozsah adries užívateľov tunelového prepojenia, ktoré sa nachádzajú za bránami.

Parametre a politika správy kľúčov:

Politiku správy kľúčov môžete prispôsobiť zadaním parametrov, ktoré sa majú použiť počas vyjednávania výmeny IKE. Existujú napríklad politiky správy kľúčov pre predzdieľaný kľúč alebo autentifikáciu podpisového režimu. Počas fázy 1 musí užívateľ určiť isté vlastnosti zabezpečenia správy kľúčov, ktoré sa použijú pri výmene.

Fáza 1 (fáza riadenia kľúčov) nastaví nasledujúce parametre konfigurácie tunela IKE:

Tunel správy kľúčov (fáza 1)

Názov tohto tunelového prepojenia IKE. Pri každom tunelovom prepojení musia byť zadané koncové body vyjednávania. Sú to dva počítače, ktoré plánujú odosielanie a overovanie platnosti správ IKE. Názov tunelového prepojenia môže popisovať koncové body tunelového prepojenia, napríklad VPN Boston alebo VPN Acme.

Host Identity Type

Typ ID, ktorý sa použije pri výmene IKE. Typ ID a hodnota sa musia zhodovať s hodnotou pre vopred zdieľaný kľúč, aby sa zaistilo vyhľadanie správneho kľúča. Ak sa na vyhľadanie hodnoty vopred zdieľaného kľúča použije samostatné ID, *ID hostiteľa* bude ID kľúča a jeho *typ* bude KEY_ID. Typ KEY_ID je užitočný v prípade, že pre jedného hostiteľa existuje viacero hodnôt vopred zdieľaných kľúčov.

Host Identity

Hodnota ID hostiteľa uvedená ako adresa IP, úplný platný názov domény (FQDN) alebo užívateľ v doméne s úplným platným názvom (*užívateľ@FQDN*). Napríklad jdoe@studentmail.ut.edu.

IP Address

Adresa IP vzdialeného hostiteľa. Táto hodnota sa požaduje v prípade, že typ ID hostiteľa je KEY_ID alebo v prípade, že typ ID hostiteľa nemožno preložiť na adresu IP. Napríklad, ak meno užívateľa nemožno rozlíšiť s lokálnym serverom názvov, bude treba zadať adresu IP pre vzdialenú stranu.

Parametre a politika správy údajov:

Parametre návrhu riadenia údajov sú nastavené počas fázy 1 konfigurácie tunela IKE. Sú rovnaké ako parametre protokolov IP Security, ktoré sa používajú pri manuálnych tunelových prepojeniach a popisujú typ ochrany slúžiaci na ochranu prenosu údajov v tunelovom prepojení. V jednom tunelovom prepojení fázy 1 môžete spustiť viacero tunelových prepojení fázy 2.

Typy údajov používajúce tunelové prepojenie IP Security popisujú nasledovné typy ID koncových bodov:

Host, Subnet alebo Range
Popisuje, či je prenos údajov tunelovým prepojením určený pre určitého hostiteľa, podsieť alebo rozsah adries.

Host/Subnet ID
Obsahuje totožnosť hostiteľa alebo podsiete lokálneho a vzdialeného systému, ktoré využívajú prenos cez toto tunelové prepojenie. Určuje ID odoslané počas vyjednávania fázy 2 a pravidlá pre filtre, ktoré sa vytvoria v prípade úspechu vyjednávania.

Subnet mask
Popisuje všetky IP adresy v podsieti (napríklad hostiteľ 9.53.250.96 a maska 255.255.255.0).

Starting IP Address Range
Poskytuje úvodnú IP adresu pre rozsah adries, ktoré budú používať tunel (napríklad 9.53.250.96 pre 9.53.250.96 až 9.53.250.93).

Ending IP Address Range
Poskytuje koncovú IP adresu pre rozsah adries, ktoré budú používať tunel (napríklad 9.53.250.93 pre 9.53.250.96 až 9.53.250.93).

Port Popisuje údaje používajúce špecifické číslo portu (napríklad 21 alebo 23).

Protocol
Popisuje údaje prenášané špecifickým protokolom (napríklad TCP alebo UDP). Určuje protokol odoslaný počas vyjednávania fázy 2 a pravidlá pre filtre, ktoré sa vytvoria v prípade úspechu vyjednávania. Protokol pre lokálny koncový bod sa musí zhodovať s protokolom pre vzdialený koncový bod.

End Port
Popisuje koncový port pre prenos údajov (napríklad 100 alebo 500). 65355 je štandardne koncovým portom.

Obmedzenie: V prípade IKEv2 použite ako selektory premávky len rozsahy adries IPv4 alebo IPv6. Koncový port sa vzťahuje len na IKEv2 a AIX 6.1 TL 04, alebo novšie.

Výber typu tunela:

Rozhodnutie, či použiť manuálne tunely alebo tunely IKE závisí od podpory tunela vzdialeného konca a typu želanej správy kľúčov.

Ak sú k dispozícii tunely IKE, použite ich, pretože poskytujú priemyselný štandard bezpečného vyjednávania o kľúčoch a ich obnove. Umožňujú aj použitie typov hlavičiek AH a IETF ESP a podpory pre ochranu anti-replay. V prípade potreby môžete nakonfigurovať režim podpisovania, ktorý umožňuje používanie digitálnych certifikátov.

Ak vzdialený koncový bod používa niektorý z algoritmov vyžadujúcich manuálne tunelové prepojenia, mali by sa používať manuálne tunelové prepojenia. Manuálne tunelové prepojenia zaisťujú interoperabilitu s veľkým množstvom hostiteľov. Keďže kľúče sú statické, ťažko sa dajú zmeniť a ich aktualizovanie nie je jednoduché, nie sú až také bezpečné. Manuálne tunelové prepojenia sa môžu použiť medzi hostiteľom s týmto operačným systémom a ľubovoľným počítačom používajúcim protokoly IP Security a rovnakú množinu algoritmov šifrovania a autentifikácie. Väčšina dodávateľov ponúka algoritmy Keyed MD5 a DES alebo HMAC MD5 a DES. Táto podmnožina spolupracuje takmer so všetkými implementáciami protokolov IP Security.

Procedúra použitá na nastavenie manuálnych tunelových prepojení závisí od toho, či nastavujete prvého hostiteľa tunelového prepojenia alebo druhého hostiteľa, ktorého parametre sa musia zhodovať s nastavením prvého hostiteľa. Pri nastavovaní prvého hostiteľa je možné kľúče generovať automaticky a algoritmy môžu byť predvolené. Pri nastavovaní druhého hostiteľa je vhodné importovať informácie o tunelovom prepojení zo vzdialeného koncového bodu (ak je to vôbec možné).

Dôležité je vedieť aj to, či sa vzdialený systém nachádza za bránou firewall. Ak je to tak, do nastavenia musia byť zahrnuté aj informácie o tejto bráne firewall.

Používanie IKE s DHCP alebo dynamicky priradenými adresami:

Jeden spoločný scenár pre používanie IP Security s operačným systémom sa uplatňuje, keď vzdialené systémy iniciujú relácie IKE so serverom a ich identita nemôže byť spojená s určitou IP adresou.

Tento prípad sa môže vyskytnúť v prostredí LAN (Local Area Network), napríklad použitie bezpečnosti IP na pripojenie k serveru na LAN a želanie zašifrovať údaje. Ostatné bežné použitia zahŕňajú vzdialených klientov vytáčajúcich server a použitie buď plne kvalifikovaného názvu domény (FQDN) alebo adresy elektronickej pošty (user@FQDN) na identifikáciu vzdialeného ID.

Vo fáze správy kľúčov (vo fáze 1) je podpis RSA jediným podporovaným režimom autentifikácie za predpokladu, že používate hlavný režim s ID, ktoré nie sú adresami IP. Inými slovami, ak chcete použiť autentifikáciu predzdieľaného kľúča, musíte použiť agresívny alebo hlavný režim s IP adresami ako ID. Ak je však počet klientov DHCP, s ktorými chcete vytvoriť tunely IPsec, veľký, definovanie jedinečných predzdieľaných kľúčov pre každého klienta DHCP bude nepraktické, preto sa v tomto scenári odporúča použitie autentifikácie podpisu RSA. V definícii tunela môžete použiť aj skupinové ID ako vzdialený ID, takže tunel budete definovať len raz pre všetkých klientov DHCP (pozrite si vzorový súbor definície tunela /usr/samples/ipsec/group_aix_responder.xml). Skupinové ID je jedinečnou funkciou AIX IPsec. Skupinové ID možno definovať tak, aby zahŕňal ľubovoľné ID IKE (ako jedna adresa IP), FQDN, užívateľské FQDN a rozsah alebo sadu adries IP, atď. a potom ho môžete použiť v definícii vášho tunela ako vzdialený ID fázy 1 alebo 2.

Poznámka: Keď sa používa skupinové ID, tunel by mal byť definovaný len ako rola odpovedača, čo znamená, že musíte aktivovať tento tunel zo strany klienta DHCP.

Keď sa pre fázu správy údajov (fáza 2) vytvárajú asociácie bezpečnosti IP na šifrovanie prevádzky TCP alebo UDP, možno nakonfigurovať generický tunel správy údajov. Ak adresa IP nie je explicitne nakonfigurovaná v databáze, každá požiadavka autentifikovaná počas fázy 1 použije všeobecné tunelové prepojenie pre definovanú fázu správy údajov. Adresy tak môžu použiť všeobecné tunelové prepojenie a môžu sa používať za predpokladu, že prísne overenie platnosti zabezpečenia na základe verejného kľúča vo fáze 1 bolo úspešné.

Použitie XML na definovanie všeobecného tunelového prepojenia na správu údajov:

Pri definovaní všeobecného tunelového prepojenia na správu údajov môžete použiť formát XML, ktorý je zrozumiteľný pre databázu **ikedb**.

Pozrite si časť s názvom “Rozhranie príkazového riadka pre konfiguráciu tunela IKE” na strane 220, kde nájdete ďalšie informácie o rozhraní IKE XML a príkaze **ikedb**. Všeobecné tunelové prepojenia na správu údajov sa používajú s protokolom DHCP. Formát XML používa názov značky IPSecTunnel. V inom kontexte sa toto prepojenie označuje aj ako *tunelové prepojenie fázy 2*. *Všeobecný tunel na správu údajov* nie je skutočný tunel, ale IPSecProtection, ktorý sa používa, ak sa prichádzajúca správa Manažmentu údajov (pod konkrétnym manažmentom kľúčov) nezhoduje so žiadnym tunelom manažmentu údajov definovaným pre tento tunel manažmentu kľúčov. Používa sa len v prípade, že respondentom je systém AIX. Zadanie všeobecného tunelového prepojenia IPSecProtection na správu údajov nie je povinné.

Všeobecné tunelové prepojenie na správu údajov je definované v prvku IKEProtection. Používajú sa na to dva atribúty XML s názvom *IKE_IPSecDefaultProtectionRef* a *IKE_IPSecDefaultAllowedTypes*.

Najprv je potrebné definovať atribút IPSecProtection, ktorý chcete použiť ako predvolený pre prípad, že neexistujú žiadne zhodné atribúty IPSecTunnels (tunelové prepojenia na správu údajov). Atribút IPSec_ProtectionName pre prvok IPSecProtection, ktorý sa má použiť ako predvolený, musí začínať reťazcom _defIPSprot_.

Teraz prejdite na prvok IKEProtection, ktorý chcete použiť s týmto predvoleným prvkom IPSecProtection. Zadajte atribút **IKE_IPSecDefaultProtectionRef**, ktorý obsahuje názov predvolenej hodnoty IPSec_Protection.

Musíte zadať aj hodnotu pre atribút **IKE_IPSecDefaultAllowedTypes** v tomto IKEProtection. Tento môže mať jednu alebo viacero nasledujúcich hodnôt (ak obsahuje viacero hodnôt, musia byť oddelené medzerami):

```
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
```

Tieto hodnoty zodpovedajú typom ID zadaným iniciátorom. Pri vyjednávaní výmeny IKE sa skutočné ID ignorujú. Zadané IPSecProtection sa používa vtedy, ak atribút **IKE_IPSecDefaultAllowedTypes** obsahuje reťazec začínajúci sa hodnotou Local_ zodpovedajúcou lokálnemu typu ID iniciátora a reťazec začínajúci sa hodnotou Remote_ zodpovedajúcou vzdialenému typu ID iniciátora. Inými slovami, v každom atribúte **IKE_IPSecDefaultAllowedTypes** musíte mať minimálne jednu hodnotu Local_ a minimálne jednu hodnotu Remote_, aby ste mohli použiť zodpovedajúci IPSec_Protection.

Príklad všeobecného tunelového prepojenia na správu údajov:

Tunelové prepojenie Data Management môžete použiť na zaslanie správy do systému.

Iniciátor zasiela do systému AIX v hlásení 2. fázy (správa údajov):

```
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
```

Systém AIX nemá vytvorené tunelové prepojenie na správu údajov, ktoré by zodpovedalo týmto ID. Má však k dispozícii prvok IPSecProtection, pre ktorý sú definované tieto atribúty:

```
IKE_IPSecDefaultProtectionRef="_defIPSprot_protection4"
IKE_IPSecDefaultAllowedTypes="Local_IPV4_Address
                              Remote_IPV4_Address
                              Remote_IPV4_Subnet
                              Remote_IPV4_Address_Range"
```

Typ ID lokálneho systému v prichádzajúcej správe (IPV4_Address) sa zhoduje s jednou z hodnôt Local_ povolených typov (Local_IPV4_Address). Aj ID vzdialeného systému v správe (IPV4_Subnet) sa zhoduje s hodnotou Remote_IPV4_Subnet. Vyjednávanie tunelového prepojenia na správu údajov bude preto pokračovať s atribútom _defIPSprot_protection4 ako s prvkom IPSecProtection.

Súbor XML /usr/samples/ipsec/default_p2_policy.xml obsahuje úplnú definíciu všeobecného prvku IPSecProtection, ktorá sa môže použiť ako príklad.

Konfigurácia tunelov Internet Key Exchange

Tunely Internet Key Exchange (IKE) môžete nakonfigurovať pomocou nástroja SMIT (System Management Interface Tool) alebo príkazového riadka.

Používanie rozhrania SMIT pre konfiguráciu tunela IKE:

Rozhranie SMIT môžete použiť na konfiguráciu tunelových prepojení IKE a vykonanie základných databázových funkcií IKE.

Nástroj SMIT používa na pridanie, odstránenie a úpravu položiek v definíciách tunelov IKE základné funkcie príkazov XML. Tento nástroj sa používa na rýchlu konfiguráciu tunelov IKE a poskytuje príklady syntaxe XML používané na vytvorenie definícií tunelov IKE. Ponuky nástroja SMIT pre IKE umožňujú vykonať zálohovanie, obnovu a inicializáciu databázy IKE.

Na konfiguráciu tunela IKE protokolu IPv4 použite rýchlu cestu **smitty ike4**. Ak chcete konfigurovať tunel IKE protokolu IPv6, použite rýchlu cestu **smitty ike6**. Funkcie databázy IKE sa nachádzajú v ponuke Advanced IP Security Configuration.

Rozhranie príkazového riadka pre konfiguráciu tunela IKE:

Príkaz **ikedb** umožňuje užívateľovi načítavať, aktualizovať, vymazávať, importovať a exportovať informácie v databáze IKE pomocou rozhrania XML.

Príkaz **ikedb** umožňuje užívateľovi zapisovať do (put) alebo čítať z (get) databázy IKE. Vstupným a výstupným formátom je súbor XML (Extensible Markup Language). Formát súboru XML je určený definíciou DTD (Document Type Definition). Príkaz **ikedb** užívateľovi umožňuje zobraziť definíciu DTD používanú na overenie súboru XML pri vykonávaní príkazu put. Pomocou príznaku **-e** možno do definície DTD pridať deklarácie entít, čo predstavuje jedinú možnú úpravu v tejto definícii. Všetky externé deklarácie DOCTYPE vo vstupnom súbore XML budú ignorované, pričom interné deklarácie DOCTYPE môžu zapríčiniť chybu. Pravidlá analýzy súboru XML pomocou definície DTD sú špecifikované v štandarde XML. Súbor /usr/samples/ipsec je typickým príkladom súboru XML, ktorý definuje bežné scenáre tunelov. Prezrite si popis príkazu **ikedb** v dokumente *Commands Reference*, kde nájdete podrobnosti syntaxe.

Príkaz **ike** možno použiť na spustenie, zastavenie a monitorovanie tunelov IKE. Príkaz **ike** možno použiť aj na aktiváciu, odstránenie alebo vypisovanie tunelov bezpečnosti IP a IKE. Prezrite si popis príkazu **ike** v dokumente *Commands Reference*, kde nájdete podrobnosti syntaxe.

Nasledovné príkazy zobrazujú spôsob použitia príkazu **ike**, **ikedb** a niekoľkých ďalších príkazov na konfiguráciu a kontrolu stavu tunela IKE:

1. Ak chcete začať vyjednávať o tuneli (*aktivovať* tunel) alebo povoliť prichádzajúcemu systému konať ako odpovedajúci (v závislosti od uvedenej roly), použite príkaz **ike** s číslom tunela:

   ```
   # ike cmd=activate numlist=1
   ```

   Môžete takisto použiť vzdialené ID alebo adresy IP, ako je uvedené v nasledovných príkladoch:

   ```
   # ike cmd=activate remid=9.3.97.256
   # ike cmd=activate ipaddr=9.3.97.100, 9.3.97.256
   ```

   Keďže dokončenie príkazov môže niekoľko sekúnd trvať, príkaz sa vracia po začatí vyjednávania.

2. Ak chcete zobraziť stav tunela, použite príkaz **ike** nasledovným spôsobom:

   ```
   # ike cmd=list
   ```

   Zobrazí sa výstup podobný nasledovnému:

   ```
   Phase 1 Tunnel ID       [1]
   Phase 2 Tunnel ID       [1]
   ```

   Výstup zobrazuje tunely fázy 1 a fázy 2, ktoré sú momentálne aktívne.

3. Ak chcete zobraziť podrobné informácie o tuneli, použite príkaz **ike** nasledovným spôsobom:

   ```
   # ike cmd=list verbose
   ```

   Výstup sa podobá tomuto:

   ```
   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. Ak chcete zobraziť pravidlá filtrovania v tabuľke dynamických filtrov pre novo aktivovaný tunel IKE, použite príkaz **lsfilt** nasledovne:

   ```
   # lsfilt -d
   ```

   Výstup sa podobá tomuto:

   ```
   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
   ```

   Tento príklad zobrazuje počítač s jedným tunelom IKE, bez ďalších tunelov. Pravidlo umiestnenia dynamického filtra (pravidlo číslo 2 v tomto vzorovom výstupe statickej tabuľky) môže užívateľ premiestniť za účelom kontroly umiestnenia, ktoré je relatívne vo vzťahu ku všetkým ostatným pravidlám definovaným užívateľom. Pravidlá v dynamickej tabuľke sú vytvárané automaticky počas vyjednávania tunelov, pričom do tabuľky filtrovania sa vkladajú zodpovedajúce pravidlá. Tieto pravidlá možno zobraziť, no nie upraviť.

5. Ak chcete zapnúť protokolovanie dynamických pravidiel filtrovania, nastavte voľbu protokolovania pre pravidlo #2 na Yes a použite príkaz **chfilt** podľa nasledujúceho príkladu:

   ```
   # chfilt -v 4 -n 2 -l y
   ```

   Podrobnosti o protokolovaní prevádzky IKE nájdete v časti “Protokolovacie zariadenia” na strane 245.

6. Ak chcete tunel deaktivovať, použite príkaz **ike** nasledovne:

   ```
   # ike cmd=remove numlist=1
   ```

7. Ak si chcete prezerať definície tunela, použite príkaz **ikedb** nasledovne:

   ```
   # ikedb -g
   ```

8. Ak chcete vložiť definície do databázy IKE zo súboru XML vygenerovaného na partnerskom počítači a prepísať existujúce objekty v databáze s tým istým názvom, použite príkaz **ikedb** týmto spôsobom:

   ```
   # ikedb -pFs peer_tunnel_conf.xml
   ```

   Súbor peer_tunnel_conf.xml je súbor XML vygenerovaný počítačom na druhej strane.
9. Ak chcete získať definíciu tunela fázy 1 s názvom *tunnel_sys1_and_sys2* a všetkých závislých tunelov fázy 2 s príslušnými návrhmi a ochranami, použite príkaz **ikedb** nasledovným spôsobom:

   ```
   # ikedb -gr -t IKETunnel -n tunnel_sys1_and_sys2
   ```
10. Ak chcete z databázy odstrániť všetky vopred zdieľané kľúče, použite príkaz **ikedb** nasledovným spôsobom:

    ```
    # ikedb -d -t IKEPresharedKey
    ```

Všeobecné informácie o podpore skupiny tunela IKE nájdete v časti “Podpora skupiny”. Príkaz **ikedb** môžete použiť na definovanie skupín z príkazového riadka.

AIX Afinita IKE a Linux:

Je možné nakonfigurovať tunelové prepojenie AIX IKE pomocou konfiguračných súborov Linux.

Ak chcete nakonfigurovať tunel AIX IKE pomocou konfiguračných súborov systému Linux, zadajte príkaz **ikedb** s príznakom **-c** (voľba konverzie), ktorý vám umožní použiť konfiguračné súbory /etc/ipsec.conf a /etc/ipsec.secrets systému Linux ako definície tunelu IKE. Príkaz **ikedb** zanalyzuje konfiguračné súbory Linux, vytvorí súbor XML a voliteľne pridá do databázy IKE definície tunela XML. Definície tunela môžete zobraziť pomocou príkazu **ikedb -g**.

Podpora skupiny:

Bezpečnosť IP podporuje zoskupovanie identifikátorov IKE v definícii tunela na priradenie viacerých ID k jednej bezpečnostnej politike bez nutnosti vytvoriť osobitné definície tunela.

Zoskupenie je užitočné najmä pri nastavovaní pripojení k viacerým vzdialeným hostiteľom, pretože nie je potrebné nastavovať alebo spravovať viaceré definície tunelov. Rovnako, v prípade potreby vykonania zmien v politike zabezpečenia nie je potrebné meniť niekoľko definícií tunelov.

Skupina musí byť definovaná pred použitím tohto názvu skupiny v definícii tunelu. Veľkosť skupiny je limitovaná na 1 KB. Na strane iniciátora vyjednávania môžete použiť skupiny ako vzdialené ID len v definíciách tunelu dátového manažmentu. Na odpovedajúcej strane vyjednávania môžete použiť skupiny ako vzdialené ID v správe kľúčov a tunelových definíciách dátového manažmentu.

Skupina pozostáva z názvu skupiny a zoznamu ID tunelov IKE a typov ID. ID môžu byť rovnakého typu alebo kombinácia nasledujúceho:

- Adresy protokolu IPv4
- Adresy protokolu IPv6
- FQDN
- user@FQDN
- Typy X500 DN

Počas vyjednávania Security Association sa v skupine lineárne vyhľadáva prvé zodpovedajúce ID.

Informácie o definovaní skupín z príkazového riadka nájdete v téme “Rozhranie príkazového riadka pre konfiguráciu tunela IKE” na strane 220.

Scenáre konfigurácie tunela IKE:

Nasledovné scenáre popisujú situácie, s ktorými sa pri nastavovaní tunelov stretáva väčšina zákazníkov. Tieto scenáre možno opísať na príklade pobočky, obchodného partnera a vzdialeného prístupu.

- V prípade pobočky má zákazník dve dôveryhodné siete, ktoré chce spolu spojiť - technickú skupinu jedného miesta s technickou skupinou iného miesta. V tomto príklade existujú dve navzájom prepojené brány, pričom prenosy medzi nimi používajú rovnaký tunel. Prenesené údaje sú na oboch koncoch tunela odpuzdrené a takto prechádzajú sieťou intranet spoločnosti.

  V prvej fáze vyjednávania IKE sa medzi dvoma bránami vytvorí priradenie zabezpečenia IKE. Údaje prechádzajúce tunelom funkcie IP Security predstavujú prenos medzi obidvoma podsieťami, pričom ID týchto podsietí sa používajú vo fáze 2 vyjednávania. Po zadaní parametrov tunela a politiky zabezpečenia pre tunel sa vytvorí číslo tunela. Na spustenie tunela použite príkaz **ike**.
- V scenári obchodného partnera nie sú siete dôveryhodné, pričom sieťoví administrátori môžu požadovať obmedzenie prístupu pre nižší počet hostiteľov za bránou zabezpečenia. V tomto prípade prenáša tunel medzi danými hostiteľmi údaje chránené funkciou IP Security používanou medzi dvoma príslušnými hostiteľmi. Protokol tunela fázy 2 je AH alebo ESP. Tento tunel medzi hostiteľmi je zabezpečený v rámci tunela medzi bránami.
- V prípade vzdialeného prístupu sú tunely nastavené na vyžiadanie, pričom sa používa vysoká úroveň zabezpečenia. Adresy IP nemusia mať význam, preto sa uprednostňujú úplné platné názvy domén, prípadne úplné platné názvy domén typu *user@*. Na vyjadrenie vzťahu kľúča a ID hostiteľa môžete použiť KEYID.

Základné pojmy o digitálnych certifikátoch a manažérovi kľúčov

Digitálne certifikáty viažu identitu na verejný kľúč, prostredníctvom ktorého môžete skontrolovať odosielateľa alebo príjemcu šifrovaného prenosu.

IP Security využíva digitálne certifikáty na povolenie *šifrovania pomocou verejných kľúčov* (nazývané aj *asymetrické šifrovanie*), ktoré šifruje údaje pomocou súkromného kľúča známeho iba užívateľovi a dešifruje ich pomocou súvisiaceho verejného (zdieľaného) kľúča z daného páru verejný - súkromný kľúč. *Páry kľúčov* sú dlhé reťazce údajov, ktoré zohrávajú úlohu kľúčov v šifrovacej schéme užívateľa.

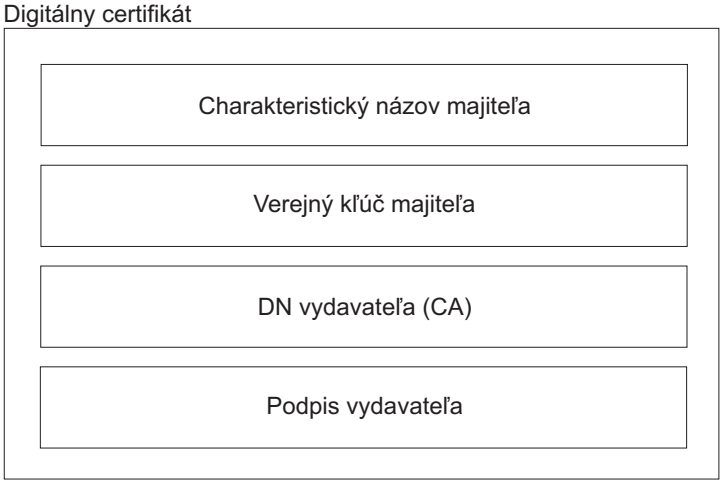
Pri kryptografii s použitím verejného kľúča sa verejný kľúč poskytuje každému, s kým chce užívateľ komunikovať. Odosielateľ digitálne podpíše zabezpečenú komunikáciu prostredníctvom zodpovedajúceho súkromného kľúča z priradeného páru kľúčov. Príjemca použije verejný kľúč na overenie podpisu odosielateľa. Ak sa správu podarí úspešne dešifrovať pomocou verejného kľúča, príjemca sa môže ubezpečiť, že ide o autentifikovaného odosielateľa.

Kryptografia verejného kľúča sa spolieha na dôveryhodné tretie strany známe ako *certifikačné autority* (*CA*) vydávajúce spoľahlivé digitálne certifikáty. Príjemca určuje, ktoré vydavateľské organizácie alebo úrady sa budú považovať za spoľahlivé. Certifikát sa vydáva na určité časové obdobie a po uplynutí dátumu ukončenia platnosti ho je nutné nahradiť.

Systém AIX poskytuje nástroj Key Manager, pomocou ktorého môžete spravovať digitálne certifikáty. Základné informácie o samotných certifikátoch sú uvedené v nasledujúcich sekciách.

Formát digitálnych certifikátov:

Digitálny certifikát obsahuje špecifické informácie o totožnosti vlastníka certifikátu a o certifikačnom úrade. Ilustrácia digitálneho certifikátu je uvedená na nasledovnom obrázku.



Obsah digitálneho certifikátu

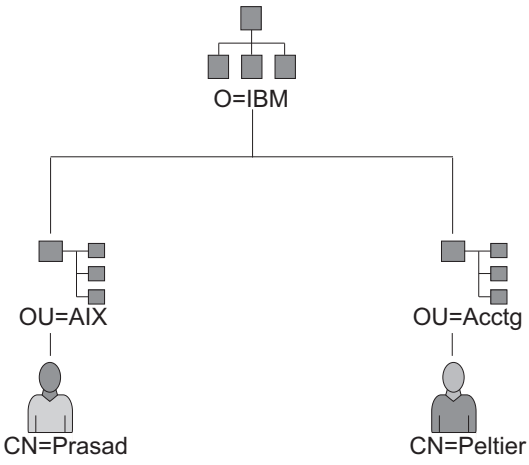
Obrázok 10. Obsah digitálneho certifikátu

Na tejto ilustrácii sú znázornené štyri entity digitálneho certifikátu. Ide o tieto entity: charakteristický názov vlastníka, verejný kľúč vlastníka, charakteristický názov vydavateľa (CÚ) a podpis vydavateľa.

Nasledujúci zoznam ďalej opisuje obsah digitálneho certifikátu:

Charakteristický názov majiteľa
Kombinácia bežného mena vlastníka a kontext (pozícia) v strome adresárov. Na nasledovnom obrázku s príkladom jednoduchého stromu adresárov je meno vlastníka Prasad a kontext je country=US, organization=ABC, a lower organization SERV, takže charakteristický názov je:

```
/C=US/O=ABC/OU=SERV/CN=prasad.austin.ibm.com
```



Príklad odvodenia charakteristického názvu z adresárového stromu

Obrázok 11. Príklad odvodenia charakteristického názvu z adresárového stromu

Na tejto ilustrácii je znázornený strom adresárov s najvyššou úrovňou O=ABC, ktorý sa na nasledujúcej úrovni rozdeľuje na dve vetvy. Druhá úroveň obsahuje OU=AIX a OU=Acctg na samostatných vetvách; každá má vetvu vedúcu k jednej entite na poslednej úrovni. Posledná úroveň obsahuje CN=Prasad a CN=Peltier v uvedenom poradí.

Verejný kľúč majiteľa
Slúži príjemcom na dešifrovanie údajov.

Alternatívny názov subjektu
Môže sa použiť identifikátor ako adresa IP, e-mailová adresa, úplný platný názov domény, atď.

Dátum vydania
Dátum vydania digitálneho certifikátu.

Dátum uplynutia platnosti
Dátum ukončenia platnosti digitálneho certifikátu.

Charakteristický názov vydavateľa
Charakteristický názov certifikačného úradu.

Digitálny podpis vydavateľa
Digitálny podpis slúžiaci na overenie platnosti certifikátu.

Úvahy o bezpečnosti digitálnych certifikátov:

Digitálny certifikát sám osebe nemôže dokázať totožnosť.

Digitálny certifikát vám umožňuje overiť totožnosť vlastníka digitálneho certifikátu až po získaní verejného kľúča, ktorý je potrebný na kontrolu digitálneho podpisu vlastníka. Svoj verejný kľúč môžete bez obáv poslať komukoľvek, pretože odoslané údaje sa nedajú dešifrovať bez vášho súkromného kľúča (druhej polovice páru kľúčov). Každý vlastník si preto musí starostlivo chrániť svoj súkromný kľúč, ktorý dopĺňa verejný kľúč v digitálnom certifikáte. Osoby, ktoré poznajú súkromný kľúč, môžu dešifrovať celú komunikáciu vlastníka digitálneho certifikátu. Bez súkromného kľúča sa digitálny certifikát nedá zneužiť.

Certifikačné autority a hierarchie dôveryhodnosti:

Digitálny certifikát je len taký dôveryhodný ako certifikačná autorita (CA), ktorá ho vydala.

Pri posudzovaní dôveryhodnosti je potrebné zohľadňovať aj politiku, ktorou sa vydávanie certifikátov riadi. Organizácie a užívatelia musia sami určiť, ktoré certifikačné úrady je možné považovať za dôveryhodné.

Nástroj Key Manager umožňuje organizáciám vytvárať certifikáty s ich podpisom, ktoré môžu byť užitočné pri testovaní alebo v prostrediach s nízkym počtom užívateľov a počítačov.

Užívateľ služby zabezpečenia potrebuje na získanie digitálnych certifikátov a overenie ich platnosti poznať verejný kľúč tejto služby. Samotné prijatie digitálneho certifikátu ešte nepotvrdzuje jeho autenticitu. Na overenie jeho autenticity potrebujete verejný kľúč certifikačného úradu, ktorý digitálny certifikát vydal. Ak nemáte overenú kópiu verejného kľúča CÚ, budete zrejme potrebovať ďalší digitálny certifikát na získanie verejného kľúča CÚ.

Zoznamy zrušených certifikátov:

Očakáva sa, že digitálny certifikát sa bude používať počas celého obdobia svojej platnosti. Podľa potreby však možno certifikát prehlásiť za neplatný ešte pred skončením jeho platnosti.

Prehlásenie certifikátu za neplatný budete potrebovať napríklad vtedy, ak niektorý zamestnanec odíde zo spoločnosti alebo ak bol súkromný kľúč certifikátu skompromitovaný. Ak chcete zrušiť platnosť certifikátu, je na to nutné upozorniť príslušný certifikačný úrad (CÚ). Certifikačný úrad pri zrušení certifikátu pridá sériové číslo neplatného certifikátu do zoznamu zrušených certifikátov.

Zoznamy zrušených certifikátov sú podpísané štruktúry údajov, ktoré sa pravidelne vydávajú a sprístupňujú vo verejnom depozitári. Získať ich možno zo serverov HTTP alebo LDAP. Každý zoznam obsahuje aktuálnu časovú známku a časovú známku nextUpdate aktualizácie. Každý zrušený certifikát v zozname je označený sériovým číslom svojho certifikátu.

Pri konfigurácii tunelového prepojenia IKE a použití digitálnych certifikátov ako spôsobu autentifikácie sa po výbere položky RSA Signature with CRL Checking môžete vždy ubezpečiť, že certifikát nebol zrušený. Ak je zapnutá funkcia CRL Checking, počas vyjednávania pri vytváraní tunelového prepojenia na správu kľúčov sa vyhľadá a prekontroluje zoznam zrušených certifikátov.

> **Poznámka:** Ak chcete používať túto vlastnosť bezpečnosti IP, váš systém musí byť nakonfigurovaný na použitie servera SOCKS (verzia 4 pre servery HTTP), servera LDAP alebo oboch. Ak viete, ktorý server SOCKS alebo LDAP sa používa na získanie CRL, môžete ich pridať do súboru /etc/isakmpd.conf.

Používanie digitálnych certifikátov v internetových aplikáciách:

Internetové aplikácie, ktoré používajú systémy kryptografie verejného kľúča, musia používať digitálne certifikáty pre získanie verejných kľúčov.

Existuje množstvo aplikácií, ktoré využívajú kryptografiu s použitím verejného kľúča, napríklad:

Virtuálne súkromné siete (VPN)
: Virtuálne súkromné siete, ktoré sa označujú aj ako *zabezpečené tunelové prepojenia*, je možné vytvoriť medzi systémami, ako sú napríklad brány firewall, na aktiváciu chránených spojení zabezpečených sietí cez nezabezpečené komunikačné prepojenia. Všetky prenosy medzi zúčastnenými systémami určené pre tieto siete sú zašifrované.

  Protokoly používané pri tunelovaní sa riadia štandardami IP Security a IKE, ktoré umožňujú zabezpečené, šifrované spojenie medzi vzdialeným klientom (napríklad zamestnancom pracujúcim doma) a zabezpečeným hostiteľom alebo sieťou.

SSL (Secure Sockets Layer)
: SSL je protokol, ktorý poskytuje ochranu údajov a integritu pre komunikáciu. Používa sa vo webových serveroch na zabezpečené pripojenia medzi webovými servermi a webovými prehliadačmi, pri použití protokolu LDAP (Lightweight Directory Access Protocol) na zabezpečené pripojenia medzi klientmi a servermi LDAP a pri službe Host-on-Demand V.2 na pripojení medzi klientom a systémom hostiteľa. SSL používa digitálne certifikáty pre výmenu kľúčov, autentifikáciu servera a voliteľne aj pre autentifikáciu klienta.

Zabezpečená elektronická pošta
: Množstvo systémov elektronickej pošty, ktoré používajú štandardy ako PEM alebo S/MIME pre zabezpečenú elektronickú poštu, používa digitálne certifikáty na digitálne podpisovanie a na výmenu kľúčov na šifrovanie a dešifrovanie poštových správ.

Digitálne certifikáty a žiadosti o certifikát:

Ak žiadate o digitálny certifikát, je nutné vytvoriť *žiadosť o certifikát* a odoslať ju certifikačnej autorite.

Podpísaný digitálny certifikát obsahuje polia pre jedinečné meno vlastníka, verejný kľúč vlastníka, jedinečný názov certifikačnej autority a podpis certifikačnej autority. Digitálny certifikát organizácie obsahuje jedinečné meno, verejný kľúč a podpis jeho vlastníka.

Žiadosti o certifikáty obsahujú polia obsahujúce jedinečné meno, verejný kľúč a podpis žiadateľa. Certifikačná autorita overí podpis žiadateľa s verejným kľúčom v digitálnom podpise s cieľom overiť, že:

- Žiadosť o certifikát nebola zmenená počas prenosu medzi žiadateľom a certifikačnou autoritou.
- Žiadateľ má príslušný súkromný kľúč pre verejný kľúč uvedený v žiadosti o certifikát.

Certifikačná autorita je v istej miere tiež zodpovedná za overenie identity žiadateľa. Požiadavky na toto overenie môžu byť v rozsahu od minimálneho overenia až po absolútne uistenie sa totožnosťou vlastníka.

Nástroj Key Manager:

Nástroj Key Manager riadi digitálne certifikáty a nachádza sa v súbore gskkm.rte nastavenom na rozširujúci balík.

Ak chcete nastaviť podporu pre digitálne certifikáty a podpisy, je nutné vykonať aspoň úlohy 1, 2, 3, 4, 6 a 7. Potom vytvorte tunel IKE a priraďte k tunelu politiku, ktorá ako metódu autentifikácie používa podpis RSA.

Databázu kľúčov môžete vytvoriť a nakonfigurovať zadaním príkazu certmgr, ktorý otvorí nástroj Key Manager.

Táto časť opisuje spôsob použitia nástroja Key Manager na tieto úlohy:

Vytvorenie databázy kľúčov:

Databáza kľúčov umožňuje prepojenie koncových bodov sietí VPN s použitím platných digitálnych certifikátov. Formát databázy kľúčov(*.kdb) sa používa s VPN protokolu IP Security.

S nástrojom Key Manager sa poskytujú tieto typy digitálnych certifikátov CA:
- Certifikačný úrad RSA Secure Server
- Certifikačný úrad Thawte Personal Premium
- Certifikačný úrad Thawte Personal Freemail
- Certifikačný úrad Thawte Personal Basic
- Certifikačný úrad Thawte Personal Server
- Certifikačný úrad Thawte Server
- Verejný primárny certifikačný úrad Verisign triedy 1
- Verejný primárny certifikačný úrad Verisign triedy 2
- Verejný primárny certifikačný úrad Verisign triedy 3
- Verejný primárny certifikačný úrad Verisign triedy 4

Tieto podpísané digitálne certifikáty umožňujú klientom pripojiť sa na servery s platnými digitálnymi certifikátmi od týchto podpisovateľov. Vytvorenú databázu kľúčov môžete v pôvodnom tvare použiť na pripojenie na server s platným digitálnym certifikátom od niektorého z týchto podpisovateľov.

Ak chcete použiť podpísaný digitálny certifikát, ktorý sa nenachádza v zozname, musíte si ho vyžiadať od certifikačného úradu a pridať ho do databázy kľúčov. Pozrite si časť “Pridanie základného digitálneho certifikátu CA” na strane 229.

Ak chcete na vytvorenie databázy kľúčov použiť príkaz **certmgr**, použite tento postup:
1. Spustite nástroj Key Manager napísaním:
   ```
   # certmgr
   ```
2. Zo zoznamu Key Database File si vyberte **New**.
3. Akceptujte predvolenú hodnotu CMS key database file v poli **Key database type**.
4. Do poľa **File Name** zadajte tento názov súboru:
   ```
   ikekey.kdb
   ```
5. Do poľa **Location** zadajte nasledovné umiestnenie databázy:
   ```
   /etc/security
   ```

   Poznámka: Databáza kľúčov musí mať názov ikekey.kbd a musí sa nachádzať v adresári /etc/security. V opačnom prípade nemôže IP Security správne fungovať.
6. Kliknite na tlačidlo **OK**. Zobrazí sa obrazovka **Password Prompt**.
7. Zadajte heslo do poľa **Password** a potom ho zadajte aj do poľa **Confirm Password**.
8. Ak chcete zmeniť počet dní platnosti hesla, zadajte požadovaný počet dní do poľa **Set expiration time?**. Predvolená hodnota v tomto poli je 60 dní. Ak nechcete, aby uplynula platnosť tohto hesla, odstráňte hodnotu z poľa **Set expiration time?**.
9. Ak chcete uložiť šifrovanú verziu hesla v súbore tajomstiev, do poľa **Stash the password to a file?** zadajte Yes.

Poznámka: Ak chcete aktivovať používanie digitálnych certifikátov s protokolmi IP Security, heslo musí byť skryté.

10. Kliknite na tlačidlo **OK**. Zobrazí sa obrazovka pre potvrdenie na overenie vytvorenia databázy kľúčov.
11. Opätovným kliknutím na **OK** sa vrátite na obrazovku IBM Key Management. Môžete vykonať ďalšie úlohy alebo ukončiť prácu s nástrojom.

Pridanie základného digitálneho certifikátu CA:

Potom, ako ste od CA žiadali a dostali základný digitálny certifikát, môžete ho pridať do svojej databázy.

Väčšina základných digitálnych certifikátov má tvar *.arm, ako napríklad:

```
cert.arm
```

Pri pridávaní základného digitálneho certifikátu CA do databázy postupujte takto:

1. Pokiaľ ešte nepoužívate nástroj Key Manager, spustite ho napísaním:
   ```
   # certmgr
   ```
2. Na hlavnej obrazovke si zo zoznamu Key Database File vyberte **Open**.
3. Zvýraznite súbor databázy kľúčov, do ktorého chcete pridať základný digitálny certifikát certifikačného úradu, a kliknite na položku **Open**.
4. Zadajte heslo a kliknite na tlačidlo **OK**. Po prijatí hesla sa vrátite na obrazovku IBM Key Management. V záhlaví okna sa zobrazuje názov vybratého súboru databázy kľúčov, čo naznačuje, že súbor je už otvorený a pripravený na používanie.
5. Vyberte si **Signer Certificates** zo zoznamu **Personal/Signer Certificates**.
6. Kliknite na tlačidlo **Add**.
7. Vyberte si typ údajov zo zoznamu **Data type**, napríklad:
   ```
   Base64-encoded ASCII data
   ```
8. Zadajte názov a umiestnenie súboru základného digitálneho certifikátu certifikačného úradu, alebo kliknite na tlačidlo **Browse** a vyberte názov a umiestnenie.
9. Kliknite na tlačidlo **OK**.
10. Zadajte štítok pre základný digitálny certifikát CA, napríklad Test CA Root Certificate, a kliknite na **OK**. Vrátite sa na obrazovku **Key Management**. V poli **Signer Certificates** sa zobrazuje návestie práve pridaného základného digitálneho certifikátu certifikačného úradu. Môžete vykonať ďalšie úlohy alebo ukončiť prácu s nástrojom.

Vytvorenie dôveryhodného nastavenia:

Nainštalované certifikáty certifikačného úradu sa na základe predvoleného nastavenia nastavia ako dôveryhodné. Podľa potreby môžete zmeniť nastavenie dôvery.

Ak chcete zmeniť dôveryhodné nastavenie, zvoľte tento postup:

1. Pokiaľ ešte nepoužívate nástroj Key Manager, spustite ho napísaním:
   ```
   # certmgr
   ```
2. Na hlavnej obrazovke si vyberte **Open** zo zoznamu **Key Database File**.
3. Zvýraznite súbor databázy kľúčov, v ktorom chcete zmeniť predvolený digitálny certifikát, a kliknite na položku **Open**.
4. Zadajte heslo a kliknite na tlačidlo **OK**. Po prijatí hesla sa vrátite na obrazovku **IBM Key Management**. V záhlaví okna sa zobrazuje názov vybratého súboru databázy kľúčov, čo naznačuje, že súbor je už otvorený.
5. Vyberte si **Signer Certificates** zo zoznamu **Personal/Signer Certificates**.
6. Zvýraznite certifikát, ktorý chcete zmeniť, a kliknite na položku **View/Edit** alebo dvakrát kliknite na záznam. Pre položku certifikátu sa zobrazí obrazovka **Key Information**.

7. Ak chcete zmeniť tento certifikát na dôveryhodný základný certifikát, vyberte si zaškrtávacie políčko vedľa **Set the certificate as a trusted root** a kliknite na tlačidlo **OK**. Ak certifikát nie je dôveryhodný, zrušte začiarknutie políčka a kliknite na tlačidlo **OK**.
8. Kliknite na **OK** z obrazovky **Signer Certificates**. Vrátite sa na obrazovku **IBM Key Management**. Môžete vykonať ďalšie úlohy alebo ukončiť prácu s nástrojom.

Vymazanie základného digitálneho certifikátu CA:

Ak už nechcete používať jeden z CA na zozname podpisových digitálnych certifikátov, musíte vymazať základný digitálny certifikát CA.

Poznámka: Ešte pred vymazaním základného digitálneho certifikátu CA vytvorte záložnú kópiu pre prípad, že ho budete chcieť neskôr opätovne vytvoriť.

Ak chcete odstrániť základný digitálny certifikát certifikačného úradu z databázy, použite nasledovný postup:

1. Pokiaľ ešte nepoužívate nástroj Key Manager, spustite ho napísaním:
   ```
   # certmgr
   ```
2. Na hlavnej obrazovke si vyberte **Open** zo zoznamu **Key Database File**.
3. Zvýraznite súbor databázy kľúčov, z ktorého chcete odstrániť základný digitálny certifikát certifikačného úradu, a kliknite na položku **Open**.
4. Zadajte heslo a kliknite na tlačidlo **OK**. Po akceptovaní hesla sa vrátite na obrazovku **Key Management**. V záhlaví okna sa zobrazuje názov vybratého súboru databázy kľúčov, čo naznačuje, že súbor je už otvorený a pripravený na úpravu.
5. Vyberte si **Signer Certificates** zo zoznamu **Personal/Signer Certificates**.
6. Zvýraznite certifikát, ktorý chcete odstrániť, a kliknite na tlačidlo **Delete**. Zobrazí sa obrazovka **Confirm**.
7. Kliknite na tlačidlo **Yes**. Vrátite sa na obrazovku **IBM Key Management**. V poli **Signer Certificates** sa už nezobrazuje návestie základného digitálneho certifikátu certifikačného úradu. Môžete vykonať ďalšie úlohy alebo ukončiť prácu s nástrojom.

Žiadosť o digitálny certifikát:

Ak chcete získať digitálny certifikát, vytvorte žiadosť pomocou nástroja Key Manager a odošlite ju certifikačnému úradu. Súbor so žiadosťou sa vytvorí vo formáte PKCS#10. CA potom overí vašu totožnosť a zašle vám digitálny certifikát.

Ak chcete požiadať o digitálny certifikát, použite nasledovný postup:

1. Pokiaľ ešte nepoužívate nástroj Key Manager, spustite ho napísaním:
   ```
   # certmgr
   ```
2. Na hlavnej obrazovke si vyberte **Open** zo zoznamu **Key Database File**.
3. Vysvieťte súbor databázy kľúčov /etc/security/ikekey.kdb, z ktorej chcete vygenerovať žiadosť a kliknite na **Open**.
4. Zadajte heslo a kliknite na tlačidlo **OK**. Po prijatí hesla sa vrátite na obrazovku **IBM Key Management**. V záhlaví okna sa zobrazuje názov vybratého súboru databázy kľúčov, čo naznačuje, že súbor je už otvorený a pripravený na úpravu.
5. Vyberte **Create > New Certificate Request**.
6. Kliknite na tlačidlo **New**.
7. Do poľa Key Label na nasledujúcej obrazovke zadajte hodnotu pre digitálny certifikát s vlastným podpisom, napríklad
   ```
   keytest.
   ```
8. Zadajte common name (predvolenou hodnotou je názov hostiteľa) a organization a potom si vyberte country. Čo sa týka ostatných polí, akceptujte predvolené hodnoty alebo vyberte nové hodnoty.

9. Definujte názov subject alternate. Voliteľnými poliami priradenými k subject alternate sú adresa elektronickej pošty, adresa IP a názov DNS. V prípade adresy IP typu tunelového prepojenia zadajte do poľa adresy IP tú adresu IP, ktorá je nakonfigurovaná v tunelovom prepojení IKE. Pre typ ID tunelového prepojenia *user@FQDN* vyplňte pole adresy elektronickej pošty. Pre typ ID tunelového prepojenia FQDN napíšte do poľa názvu DNS plne kvalifikovaný názov domény (napríklad *hostname.companyname*.com).
10. V spodnej časti obrazovky zadajte názov pre súbor, napríklad:

    ```
    certreq.arm.
    ```
11. Kliknite na tlačidlo **OK**. Zobrazí sa obrazovka pre potvrdenie na overenie vytvorenia žiadosti o nový digitálny certifikát.
12. Kliknite na tlačidlo **OK**. Vrátite sa na obrazovku **IBM Key Management**. V poli **Personal Certificate Requests** sa zobrazuje návestie kľúča vytvorenej žiadosti o nový digitálny certifikát (PKCS#10).
13. Odošlite súbor so žiadosťou o nový digitálny certifikát certifikačnému úradu. Môžete vykonať ďalšie úlohy alebo ukončiť prácu s nástrojom.

Pridanie (Prijatie) nového digitálneho certifikátu:

Po prijatí nového digitálneho certifikátu od CA ho musíte pridať do databázy kľúčov, z ktorej ste vygenerovali požiadavku.

Pri pridávaní (prijímaní) nového digitálneho certifikátu použite túto procedúru:

1. Pokiaľ ešte nepoužívate nástroj Key Manager, spustite ho napísaním:

    ```
    # certmgr
    ```
2. Na hlavnej obrazovke si vyberte **Open** zo zoznamu **Key Database File**.
3. Zvýraznite súbor databázy kľúčov, ktorý ste použili na vytvorenie žiadosti o certifikát, a kliknite na položku **Open**.
4. Zadajte heslo a kliknite na tlačidlo **OK**. Po prijatí hesla sa vrátite na obrazovku IBM Key Management. V záhlaví okna sa zobrazuje názov vybratého súboru databázy kľúčov, čo naznačuje, že súbor je už otvorený a pripravený na úpravu.
5. Vyberte si **Personal Certificate Requests** zo zoznamu **Personal/Signer Certificates**.
6. Kliknutím na **Receive** pridajte novo prijatý digitálny certifikát do vašej databázy.
7. Vyberte si typ údajov nového digitálneho certifikátu zo zoznamu **Data type**. Predvolená hodnota je **Base64-encoded ASCII data**.
8. Zadajte názov a umiestnenie súboru certifikátu pre nový digitálny certifikát alebo kliknite na tlačidlo **Browse** a vyberte názov a umiestnenie.
9. Kliknite na tlačidlo **OK**.
10. Zadajte pre nový digitálny certifikát opisný štítok, napríklad:

    ```
    VPN Branch Certificate.
    ```
11. Kliknite na tlačidlo **OK**. Vrátite sa na obrazovku **IBM Key Management**. V poli **Personal Certificates** sa zobrazuje návestie práve pridaného nového digitálneho certifikátu. Môžete vykonať ďalšie úlohy alebo ukončiť prácu s nástrojom. Ak sa pri zavádzaní certifikátu vyskytne chyba, skontrolujte, či súbor certifikátu začína textom ——BEGIN CERTIFICATE—— a končí textom ——END CERTIFICATE——.

    Napríklad:

    ```
    -----BEGIN CERTIFICATE-----
    ajdkfjaldfwwwwwwwwwwadafdw
    kajf;kdsajkflasasfkjafdaff
    akdjf;ldasjkf;safdfdasfdas
    kaj;fdljk98dafdas43adfadfa
    -----END CERTIFICATE-----
    ```

    Ak sa text nezhoduje, upravte súbor certifikátu tak, aby mal zodpovedajúci začiatok a koniec.

Vymazanie digitálneho certifikátu:

Niekedy sa môže stať nutnosťou vymazať digitálny certifikát.

Poznámka: Skôr, než odstránite digitálny certifikát, vytvorte jeho záložnú kópiu pre prípad, že by ste ho v budúcnosti chceli obnoviť.

Pri vymazávaní digitálneho certifikátu z databázy postupujte takto:

1. Pokiaľ ešte nepoužívate nástroj Key Manager, spustite ho napísaním:
   ```
   # certmgr
   ```
2. Na hlavnej obrazovke si vyberte **Open** zo zoznamu **Key Database File**.
3. Zvýraznite súbor databázy kľúčov, z ktorého chcete odstrániť digitálny certifikát, a kliknite na položku **Open**.
4. Zadajte heslo a kliknite na tlačidlo **OK**. Po prijatí hesla sa vrátite na obrazovku **IBM Key Management**. V záhlaví okna sa zobrazuje názov vybratého súboru databázy kľúčov, čo naznačuje, že súbor je už otvorený a pripravený na úpravu.
5. Vyberte si **Personal Certificate Requests** zo zoznamu **Personal/Signer Certificates**.
6. Zvýraznite digitálny certifikát, ktorý chcete odstrániť, a kliknite na tlačidlo **Delete**. Zobrazí sa obrazovka **Confirm**.
7. Kliknite na tlačidlo **Yes**. Vrátite sa na obrazovku **IBM Key Management**. V poli **Personal Certificates** sa už nezobrazuje návestie odstráneného digitálneho certifikátu. Môžete vykonať ďalšie úlohy alebo ukončiť prácu s nástrojom.

Zmena hesla databázy:

Niekedy je potrebné zmeniť heslo databázy.

Ak chcete zmeniť heslo pre databázu kľúčov, použite nasledovný postup:

1. Pokiaľ ešte nepoužívate nástroj Key Manager, spustite ho napísaním:
   ```
   # certmgr
   ```
2. Na hlavnej obrazovke si vyberte **Change Password** zo zoznamu **Key Database File**.
3. Zadajte nové heslo do poľa **Password** a potom ho zadajte aj do poľa **Confirm Password**.
4. Ak chcete zmeniť počet dní platnosti hesla, zadajte požadovaný počet dní do poľa **Set expiration time?**. Predvolená hodnota v tomto poli je 60 dní. Ak nechcete, aby uplynula platnosť tohto hesla, odstráňte hodnotu z poľa **Set expiration time?**.
5. Ak chcete uložiť šifrovanú verziu hesla v súbore tajomstiev, do poľa **Stash the password to a file?** zadajte Yes.

   Poznámka: Ak chcete aktivovať používanie digitálnych certifikátov s protokolmi IP Security, heslo musí byť skryté.
6. Kliknite na tlačidlo **OK**. Informácia o úspešnom vykonaní požiadavky sa zobrazí v hlásení na stavovej lište.
7. Opätovným kliknutím na tlačidlo **OK** sa vrátite na obrazovku **IBM Key Management**. Môžete vykonať ďalšie úlohy alebo ukončiť prácu s nástrojom.

Vytvorenie tunelov IKE pomocou digitálnych certifikátov:

Ak chcete vytvoriť tunely IKE používajúce digitálne certifikáty, v súbore politiky transformácie tunelu IKE musíte ako režim autentifikácie nastaviť podpisy RSA.

Nasledujúci príklad predstavuje príklad súboru politiky XML určujúci podpisy RSA:

```
<!-- define the policy for IKE tunnel -->
<IKEProtection
 IKE ProtectionName="ike_3des_sha">
 <IKETTransform
  IKE AuthenticationMethod="RSA_signatures"
```

```
  IKE Encryption="3DES-CBC"
  IKE Hash="SHA"
  IKE DHGroup="1"/>
</IKEProtection>
```

IP Security podporuje nasledujúce typy hostiteľských identít tunelu IKE:

- IP addreas
- Fully Qualified Domain Name (FQDN),
- Adresa typu *meno@FQDN*.
- X.500 Distinguished Name,
- Key identifier

Keď tunel IKE používa režim podpisov RS, v definícii tunelu IKE sa zvyčajne používajú jedinečné názvy X.500. Napríklad, ak sú lokálny a vzdialený hostiteľ tunelu určený ako **/C=US/O=ABC/OU=SERV/CN=localname.austin.ibm.com** a **/C=US/O=ABC/OU=SERV/CN=remotename.austin.ibm.com**, definícia tunelu IKE v súbore XML bude podobná ako v nasledujúcom príklade:

```
<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>
```

Vyžadovaný certifikát môžete od certifikačnej autority získať vygenerovaním certifikačnej požiadavky pomocou nástroja Key Manager. Ak, napríklad, ako jedinečný názov subjektu vo svojom certifikáte používate **/C=US/O=ABC/OU=SERV/CN=name.austin.ibm.com**, v nástroji Key Manager musíte pri vytváraní žiadosti o digitálny certifikát uviesť nasledujúce hodnoty:

Common name
name.austin.ibm.com

Organization
ABC

Organizational unit
SERV

Country
US

Zadaný jedinečný názov X.500 predstavuje názov, ktoré pre vás zvyčajne nakonfiguruje administrátor systému alebo adresára LDAP. Hodnota organizačnej jednotky je dobrovoľná.

IP Security podporuje aj zadávanie iných typov identít v digitálnom certifikáte ako alternatívne názvy subjektu. Napríklad, ak ako alternatívnu identitu hostiteľa používate IP adresu 10.10.10.1, v žiadosti o digitálny certifikát musíte uviesť nasledujúce hodnoty:

Common name
name.austin.ibm.com

Organization
ABC

Organizational unit
SERV

Country
US

Subject alternate IP address field
10.10.10.1

Po vytvorení žiadosti o digitálny certifikát s týmito informáciami certifikačný úrad použije tieto informácie pri vytváraní osobného digitálneho certifikátu.

Pri požiadaní o osobný digitálny certifikát bude certifikačný úrad požadovať tieto informácie:

- Žiadate o certifikát X.509.
- Formát podpisu je MD5 so šifrovaním RSA.
- Informácie o prípadnom zadaní alternatívneho názvu subjektu. Alternatívne typy názvov sú uvedené v nasledujúcom zozname:
  - IP addreas
  - Úplný názov domény (FQDN),
  - Adresa typu *meno@FQDN*.

  Súbor so žiadosťou o certifikát obsahuje nasledovné informácie o alternatívnom názve subjektu.
- Plánované použitie kľúčov (bit pre digitálny podpis musí byť vybratý).
- Súbor žiadosti o digitálny certifikát Key Manager (vo formáte PKCS#10).

Podrobné pokyny na vytvorenie žiadosti o certifikát v nástroji Key Manager nájdete v téme “Žiadosť o digitálny certifikát” na strane 230.

Pred aktivovaním tunel IKE musíte pridať osobný digitálny certifikát, ktorý ste dostali od certifikačnej autority, do databázy nástroja Key Manager (ikekey.kdb). Viac informácií nájdete v časti “Pridanie (Prijatie) nového digitálneho certifikátu” na strane 231.

Protokol IP Security podporuje tieto typy osobných digitálnych certifikátov:

DN subjektu
Charakteristický názov subjektu musí byť zadaný v nasledovnom formáte a poradí:

`/C=US/O=ABC/OU=SERV/CN=`*meno*`.austin.ibm.com`

Nástroj Key Manager povoľuje len jednu hodnotu **OU**.

Charakteristický názov subjektu a alternatívny názov subjektu ako IP adresa
Charakteristický názov subjektu a alternatívny názov subjektu môže byť zadaný ako adresa IP, ako je to uvedené v nasledovnom príklade:

/C=US/O=ABC/OU=SERV/CN=meno.austin.ibm.com a 10.10.10.1

Charakteristický názov subjektu a alternatívny názov subjektu ako FQDN
Charakteristický názov subjektu a alternatívny názov subjektu je možné zadať ako úplný platný názov domény, ako je uvedené v nasledovnom príklade:

/C=US/O=ABC/OU=SERV/CN=name.austin.ibm.com and bell.austin.ibm.com.

Charakteristický názov subjektu a alternatívny názov subjektu ako *user@FQDN*
Charakteristický názov subjektu a alternatívny názov subjektu je možné zadať ako adresu užívateľa (*ID_užívateľa@úplný_platný_názov_domény*), ako je uvedené v nasledovnom príklade:

/C=US/O=ABC/OU=SERV/CN=meno.austin.ibm.com a meno@austin.ibm.com.

Charakteristický názov subjektu a viacero alternatívnych názvov subjektu

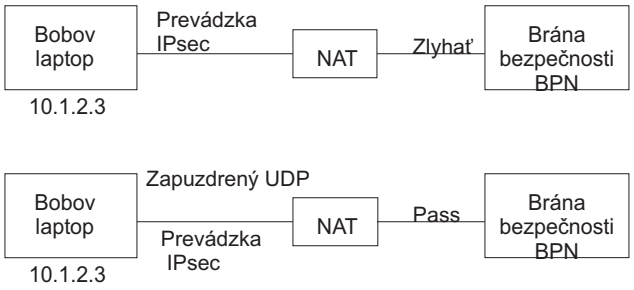
K charakteristickému názvu subjektu môže byť priradených viacero alternatívnych názvov subjektu, ako je to uvedené v nasledovnom príklade:

/C=US/O=ABC/OU=SERV/CN=name.austin.ibm.com and bell.austin.ibm.com, 10.10.10.1, and user@name.austin.ibm.com.

Preklad sieťovej adresy

IP Security používa zariadenia, ktorých adresy podliehajú prekladu sieťovej adresy (NAT).

NAT sa často používa ako súčasť technológie brány firewall na zdieľanie internetového prepojenia a ide o štandardnú funkciu na smerovačoch a okrajových zariadeniach. Protokol IP bezpečnosti závisí od identifikácie vzdialených koncových bodov a ich politiky založenej na vzdialenej IP adrese. Keď pomocné zariadenia, napríklad smerovače a firewally, prekladajú súkromnú adresu na verejnú, požadované spracovanie autentifikácie v bezpečnosti IP môže zlyhať, pretože adresa v pakete IP bola po vypočítaní súhrnu autentifikácie zmenená. S podporou NAT bezpečnosti IP sú zariadenia nakonfigurované za uzlom, ktorý vykonáva preklad sieťovej adresy, schopné vytvoriť tunel bezpečnosti IP. Kód bezpečnosti IP je schopný zistiť, kedy bola vzdialená adresa preložená. Použitie novej implementácie bezpečnosti IP s podporou pre NAT umožňuje klientovi VPN pripojiť sa z domu alebo na ceste do práce prostredníctvom internetového pripojenia s povolením NAT.



Obrázok 12. Bezpečnosť IP s povolením NAT

Tento diagram zobrazuje rozdiel medzi implementáciou bezpečnosti IP s povolením NAT so zabalenou prevádzkou UDP a implementáciou bez povolenia NAT.

Konfigurácia zabezpečenia IP pre prácu s NAT:

Aby bolo možné použiť NAT v IP Security, musíte nastaviť premennú *ENABLE_IPSEC_NAT_TRAVERSAL* v súbore /etc/isakmpd.conf. Ak je táto premenná nastavená, sú pridané pravidlá filtrovania pre posielanie a príjem premávky na porte 4500.

Nasledujúci príklad zobrazuje pravidlá filtrovania, keď je nastavená premenná *ENABLE_IPSEC_NAT_TRAVERSAL*.

```
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
```

Nastavenie premennej *ENABLE_IPSEC_NAT_TRAVERSAL* pridá do tabuľky filtrov aj niektoré ďalšie pravidlá filtrovania. Špeciálne správy IPSEC NAT používajú zabalenie UDP a aby mohla prevádzka prebiehať, musia byť pridané pravidlá filtrovania. Okrem toho sa vo fáze 1 vyžaduje podpisový režim. Ak sa adresa IP používa ako identifikátor v certifikáte, tento by mal obsahovať súkromnú IP adresu.

IP bezpečnosť tiež vyžaduje zaslať NAT správy typu keep-alive na údržbu mapovania pôvodnej IP adresy a adresy NAT. Interval je zadaný premennou *NAT_KEEPALIVE_INTERVAL* v súbore /etc/isakmpd.conf. Táto premenná uvádza v sekundách, ako často sa odosielajú pakety NAT typu keep-alive. Ak neuvediete hodnotu pre *NAT_KEEPALIVE_INTERVAL*, použije sa 20 sekúnd ako predvolená hodnota.

Obmedzenia pri používaní výmen NAT:

Koncové body za zariadeniami NAT musia chrániť svoju premávku použitím protokolu ESP.

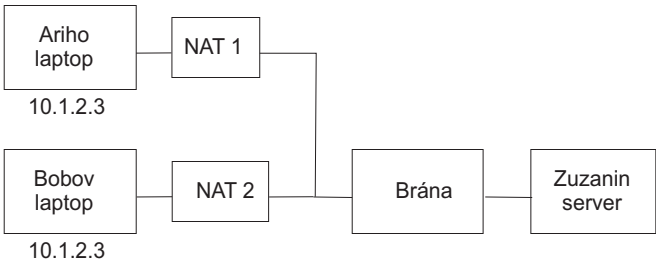
ESP je prevládajúca hlavička vybratá pre bezpečnosť IP a bude užitočná pre mnohé zákaznícke aplikácie. ESP zahŕňa hašovanie užívateľských údajov, ale nie hlavičky IP. Kontrola integrity v hlavičke AH zahŕňa adresy IP zdroja a cieľa v kontrole integrity zakľúčovanej správy. NAT alebo reverzné zariadenia NAT, ktoré vykonávajú zmeny v poliach adresy, robia kontrolu integrity správ neplatnou. Z toho dôvodu, ak je vo fáze 2 politiky pre tunel definovaný len protokol AH a vo fáze 1 výmeny sa zistí NAT, zašle sa upozornenie s oznámením NO_PROPOSAL_CHOSEN.

Okrem toho si musí pripojenie používajúce NAT vybrať režim tunelu, aby bola pôvodná IP adresa zabalená do paketu. Režim transportu a adresy s NAT nie sú kompatibilné. Ak je zistený NAT a vo fáze 2 je navrhnutý len režim transportu, zašle sa upozornenie s oznámením NO_PROPOSAL_CHOSEN.

Vyhnutie sa konfliktom tunelového režimu:

Vzdialení partneri môžu dohodnúť položky, ktoré sa v bráne prekrývajú. Takéto prekrývanie spôsobí konflikt tunelového režimu.

Nasledujúci obrázok ukazuje konflikt tunelového režimu.



Obrázok 13. Konflikt režimu tunela

Brána má dve možné priradenia zabezpečenia (SA) pre IP adresu 10.1.2.3. Tieto duplicitné vzdialené adresy spôsobujú zmätok v tom, kam poslať pakety prichádzajúce zo servera. Keď sa medzi Zuzaniným serverom a Ariho laptopom konfiguruje server, použije sa IP adresa a Zuzana si nemôže nakonfigurovať tunel s rovnakou adresou, akú má Bob. Ak

sa chcete vyvarovať konfliktu s režimom tunela, mali by ste zadefinovať tunel s rovnakou IP adresou. Keďže vzdialená adresa nie je pod kontrolou vzdialeného užívateľa, mali by byť použité iné typy ID pre identifikáciu vzdialeného hostiteľa, napríklad presne zadaný názov domény alebo užívateľ pri presne zadanom názve domény.

Konfigurácia manuálnych tunelov

V prípade, že zariadenia nepodporujú metódu automatických kľúčov, môžete nakonfigurovať manuálne tunely IP Security.

Manuálne tunely a filtre:

Proces nastavenia tunela pozostáva z definovania tunela na jednom konci, importovania danej definície na druhý koniec a aktivácie tunela a pravidiel filtrovania na oboch koncoch. Tunel je po vykonaní týchto krokov pripravený na použitie.

Pre nastavenie manuálneho tunela nie je potrebné oddelene konfigurovať pravidlá filtrovania. Ak všetky prenosy medzi dvoma hostiteľmi prechádzajú tunelom, potrebné pravidlá filtrovania sa generujú automaticky.

Ak informácie o tuneli nie sú explicitne dodané, je potrebné zabezpečiť, aby sa zhodovali na oboch stranách. Napríklad, algoritmy šifrovania a autentifikácie zadané pre zdroj budú použité pre cieľ, ak nie sú zadané cieľové hodnoty.

Vytvorenie manuálneho tunela na prvom hostiteľovi:

Tunel môžete nakonfigurovať pomocou rýchlej cesty ips4_basic nástroja SMIT (pre IPv4), rýchlej cesty ips6_basic nástroja SMIT (pre IPv6) alebo môžete tunel vytvoriť manuálne vykonaním nasledujúceho postupu.

Nižšie je uvedený príklad príkazu **gentun**, ktorý sa používa na vytvorenie manuálneho tunela:

```
gentun -v 4 -t manual -s 5.5.5.19 -d 5.5.5.8 \
   -a HMAC_MD5 -e DES_CBC_8 -N 23567
```

Na zobrazenie zoznamu vlastností manuálneho tunela vytvoreného v predchádzajúcom príklade môžete použiť príkaz **lstun -v 4**. Výstup sa podobá tomuto:

```
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    : -
```

Ak chcete aktivovať tunel, napíšte tento kód:

```
mktun -v 4 -t1
```

Pravidlá filtrovania priradené k tunelu sa vygenerujú automaticky.

Na zobrazenie pravidiel filtrovania použite príkaz **lsfilt -v 4**. Výstup sa podobá tomuto:

```
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
```

Ak chcete aktivovať pravidlá filtrovania vrátane predvolených filtrovacích pravidiel, použite príkaz **mktun -v 4 -t 1**.

Ak chcete nastaviť druhú stranu (v prípade, že ide o iný počítač používajúci tento operačný systém), definície tunela môžete exportovať na hostiteľskom počítači A importovať ich na hostiteľskom počítači B.

Nasledujúci príkaz exportuje definíciu tunela do súboru s názvom **ipsec_tun_manu.exp** a všetky priradené pravidlá filtrovania do súboru **ipsec_fltr_rule.exp** v adresári označenom príznakom **-f**:

```
exptun -v 4 -t 1 -f /tmp
```

Vytvorenie manuálneho tunela na druhom hostiteľovi:

Ak chcete vytvoriť zhodný koniec tunela, exportné súbory sa skopírujú a naimportujú na vzdialený počítač.

Na vytvorenie zhodného konca tunela použite tento príkaz:

```
imptun -v 4 -t 1 -f /tmp
```

kde

1 je tunel, ktorý sa má importovať

/tmp je adresár so súbormi importu

Číslo tunela generuje systém. Toto číslo môžete získať z výstupu príkazu **gentun** alebo použitím príkazu **lstun**, ktorý umožňuje zobraziť zoznam tunelov a určiť správne číslo tunela na import. Ak súbor importu obsahuje len jeden tunel alebo ak sa majú importovať všetky tunely, voľba **-t** nie je potrebná.

Ak na vzdialenom počítači nie je nainštalovaný tento operačný systém, súbor exportu možno použiť ako referenciu pre nastavenie algoritmu, kľúčov a hodnôt indexu parametrov zabezpečenia (SPI) pre druhý koniec tunela.

Ak chcete vytvoriť tunely, môžete importovať aj súbory exportu z produktu s bránou firewall. Pri importovaní daného súboru použite voľbu **-n** nasledovným spôsobom:

```
imptun -v 4 -f /tmp -n
```

Odstránenie filtrov:

Ak chcete úplne odstrániť filtre a zastaviť bezpečnosť IP, použite príkaz **rmdev**.

Predvolené pravidlo filtrovania je stále aktívne, aj keď je filtrovanie vypnuté s príkazom **mkfilt -d**. Tento príkaz umožňuje pozastaviť alebo odstrániť všetky pravidlá filtrovania a zaviesť nové pravidlá, kým ochrana predvoleného pravidla zostáva. Predvolené pravidlo filtrovania je *DENY*. Ak deaktivujete filtrovanie s príkazom **mkfilt -d**, správy z príkazu **lsfilt** zobrazia, že filtrovanie je vypnuté, ale žiadne pakety nemôžu dovnútra ani von. Ak chcete bezpečnosť IP zastaviť úplne, použite príkaz **rmdev**.

Konfigurácia filtrovania bezpečnosti IP

Filtrovanie možno nastaviť ako jednoduché, používajúce väčšinou automaticky generované pravidlá filtrovania, prípadne ho možno prispôsobiť definovaním veľmi špecifických funkcií filtrovania na základe vlastností paketov IP.

Každý riadok v tabuľke filtrovania sa nazýva *pravidlo*. Súhrn pravidiel určuje, ktoré pakety sú akceptované v smere do počítača a naopak a spôsob ich smerovania. Zhody s pravidlami filtrovania v prichádzajúcich paketoch sa dosahujú porovnávaním zdrojovej adresy a hodnoty SPI s údajmi v tabuľke filtra. Táto časť musí byť preto jedinečná. Pravidlá filtrovania môžu riadiť mnohé aspekty komunikácie, vrátane zdrojových a cieľových adries a masiek, protokolu, čísla portu, smeru, riadenia fragmentov, zdrojového smerovania, tunelu a typu rozhrania.

Typy pravidiel filtrovania sú nasledovné:

- Statické pravidlá filtrovania sa vytvárajú v tabuľke filtrovania používanej na všeobecné filtrovanie prevádzky alebo na priraďovanie k manuálnym tunelovým prepojeniam. Tieto pravidlá možno pridávať, odstraňovať, upravovať alebo premiestňovať. Za účelom identifikácie určitého pravidla možno pridať voliteľné textové pole s popisom.
- Automaticky generované a užívateľom zadané pravidlá filtrovania (nazývané aj *autogenerované* pravidlá filtrovania) tvoria špecifickú sadu pravidiel vytvorených na použitie tunelových prepojení IKE. Na základe informácií a vyjednávania tunela pre manažment údajov sa vytvárajú statické aj dynamické pravidlá filtrovania.
- Preddefinované pravidlá filtrovania sú generické pravidlá filtrovania, ktoré nie je možné modifikovať, presúvať ani vymazávať, napríklad pravidlo all traffic, pravidlo ah a pravidlo esp. Vzťahujú sa na všetky prenosy.

Príznak smeru (**-w**) príkazu **genfilt** sa používa, ak chcete uviesť, či sa má zadané pravidlo použiť počas vstupného alebo výstupného spracúvania paketu. Ak sa pre tento príznak použije hodnota **both**, táto zadáva, že uvedené pravidlo sa použije počas vstupného aj výstupného spracúvania. Ak je v AIX IPsec povolené filtrovanie, o osude všetkých sieťových paketov rozhoduje aspoň jedno pravidlo (bez ohľadu na to, či ide o prichádzajúci alebo odchádzajúci paket). Ak chcete použiť pravidlo len počas spracúvania prichádzajúceho (alebo odchádzajúceho) paketu, môžete tak urobiť pomocou prepínača **-w** príkazu **genfilt**. Napríklad, ak sa paket odošle z hostiteľa A na hostiteľa B, odchádzajúci IP paket má zdrojovú adresu *A* a cieľovú adresu *B*. Na hostiteľovi A sa tento paket spracuje filtrom IPsec počas spracovania odchádzajúcich údajov, kým na hostiteľovi B sa spracuje počas spracovania prichádzajúcich údajov. Predpokladajme, že sa medzi hostiteľom A a hostiteľom B nachádza brána G. V bráne G sa tento paket (všetky nemenné polia majú rovnakú hodnotu) sa spracuje dvakrát: prvýkrát počas spracovania prichádzajúcich údajov a druhýkrát počas spracovania odchádzajúcich údajov (ak je nastavená voľba **ipforwarding**). Pre paket zasielaný z hostiteľa A hostiteľovi B cez bránu G potrebujete povoľovacie pravidlo s:

- Na hostiteľovi A - **src addr** nastavené na A, **dest addr** na B, smerom von
- Na hostiteľovi B - **src addr** nastavené na A, **dest addr** na B, smerom dnu

Ale v bráne G budete potrebovať dve pravidlá:

1. **src addr** nastavené na A, **dest addr** na B, smerom von

2. **src addr** nastavené na A, **dest addr** na B, smerom dnu

Vyššie uvedené pravidlá môžete nahradiť: **src addr** nastaveným na A, **dest addr** na B a na oba smery. Hodnota **both** pre smer sa preto bežne používa v bránach, ktoré majú voľbu **ipforwarding** nastavenú na hodnotu no. Uvedená konfigurácia je len pre balíky zasielané z hostiteľa A hostiteľovi B cez bránu G. Ak chcete, aby boli balíky zasielané v opačnom smere (z hostiteľa B hostiteľovi A cez bránu G), potrebujete ďalšie pravidlo.

Poznámka: Smer **both** znamená, že priradené pravidlo sa použije pre prichádzajúce aj odchádzajúce pakety. Neznamená to však, že pravidlo sa použije, ak sú zdrojová a cieľová adresa obrátené. Napríklad, ak má server A pravidlo s *A* ako zdrojovou adresou a *B* ako cieľovou adresou a smer je nastavený na **both**, potom *A* ako prichádzajúci paket s *B* ako zdrojovou adresou a *A* ako cieľom nevyhovujú tomuto pravidlu. Voľba **both** sa zvyčajne používa v bránach, ktoré preposielajú pakety.

K týmto pravidlám filtrovania sú priradené masky podsiete, ktoré zoskupujú ID skupín priradené k pravidlu filtrovania a voľba konfigurácie typu hostiteľ-firewall-hostiteľ. Nasledovné časti popisujú rôzne typy pravidiel filtrovania a k nim priradené funkcie.

Filtre protokolu IP pre AIX:

IPFilter je softvérový balík, ktorý poskytuje preklad sieťových adries (NAT) alebo služby firewallu.

Softvér s otvoreným zdrojovým kódom IPFilter, verzia 4.1.13, bol portovaný pre systém AIX v súlade s licenciou uverejnenou na webových stránkach softvéru IPFilter (http://coombs.anu.edu.au/~avalon/). Softvér IPFilter sa poskytuje na disku AIX Expansion Pack. Inštalačný balík (ipfl) obsahuje stránku manuálu a licenciu.

V operačnom systéme AIX softvér IPFilter zavedie rozšírenie jadra /usr/lib/drivers/ipf. Binárne súbory **ipf**, **ipfs**, **ipfstat**, **ipmon** a **ipnat** sú tiež dodané spolu s balíkom.

Po nainštalovaní balíka spustite nasledujúci príkaz, čím zavediete rozšírenie kernelu:

```
/usr/lib/methods/cfg_ipf -l
```

Nasledujúcim príkazom uvoľníte rozšírenie kernelu:

```
/usr/lib/methods/cfg_ipf -u
```

Nezabudnite povoliť sieťovú voľbu ipforwarding, ak je potrebné preposielanie paketov. Viac informácií o produkte IPFilter, vrátane stránok man a často kladených otázok, si vyhľadajte na stránkach IPFilter (http://coombs.anu.edu.au/~avalon/).

Pravidlá statického filtrovania:

Každé pravidlo statického filtrovania obsahuje polia oddelené medzerami.

Nasledujúci zoznam poskytuje názov každého poľa v pravidle statického filtrovania, za ktorým nasleduje príklad z pravidla 1 v zátvorkách:

- 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).

Príklad pravidiel statického filtrovania

```
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 outbound traffic

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
```

Popis pravidiel v predchádzajúcom príklade je nasledovný:

Pravidlo 1
: Pre démon **Session Key**. Toto pravidlo sa nachádza len v tabuľkách filtrovania protokolu IP Verzia 4. Na kontrolu paketov pre obnovenie kľúča relácie používa port číslo 4001. Pravidlo 1 je príkladom, ako možno použiť číslo portu na špecifický účel.

  Poznámka: Neupravujte toto pravidlo filtrovania, s výnimkou účelov protokolovania.

Pravidlá 2 a 3
: Umožňujú spracovanie hlavičiek autentifikácie (AH) a hlavičiek ESP (encapsulating security payload).

  Poznámka: Neupravujte pravidlá 2 a 3, s výnimkou účelov protokolovania.

Pravidlá 4 a 5
: Sada automaticky generovaných pravidiel, ktoré filtrujú premávku medzi adresami 10.0.0.1 a 10.0.0.2 cez tunel 1. Pravidlo 4 je pre odchádzajúcu premávku a pravidlo 5 je pre prichádzajúcu premávku.

  Poznámka: Pravidlo 4 obsahuje užívateľom definovaný opis *odchádzajúcich prenosov*.

Pravidlá 6 až 9
: Sada užívateľom definovaných pravidiel, ktoré filtrujú odchádzajúce služby rsh, rcp, rdump, rrestore, a rdist medzi adresami 10.0.0.1 a 10.0.0.3 cez tunel 2. V tomto príklade je protokolovanie nastavené na Yes, takže administrátor môže monitorovať tento typ premávky.

Pravidlá 10 a 11
: Sada užívateľom definovaných pravidiel, ktoré filtrujú prichádzajúce i odchádzajúce služby icmp ľubovoľného typu medzi adresami 10.0.0.1 a 10.0.0.4 cez tunel 3.

Pravidlá 12 až 17
: Užívateľom definované pravidlá filtrovania, ktoré filtrujú prenosy služby FTP s adresami od 10.0.0.1 a 10.0.0.5 odchádzajúce cez tunel 4.

Pravidlo 18
: Automaticky generované pravidlo, ktoré je vždy umiestnené na koniec tabuľky. V tomto príklade povoľuje všetky pakety, ktoré nevyhovujú ostatným pravidlám filtrovania. Možno ho nastaviť tak, aby zakázalo všetky prenosy, ktoré nevyhovujú ostatným pravidlám.

Každé pravidlo možno zobraziť oddelene (pomocou príkazu **lsfilt**), čím sa zobrazí všetky polia so svojimi hodnotami. Napríklad:

```
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
```

Nasledovný zoznam obsahuje všetky parametre, ktoré v pravidle filtrovania možno zadať:

-v Verzia IP: 4 alebo 6.

-a Akcia:

- **d** Zakázať
- **p** Povoliť

-s Zdrojová adresa. Môže byť vyjadrená adresou IP alebo názvom hostiteľa.

-m Zdrojová maska podsiete.

-d Cieľová adresa. Môže byť vyjadrená adresou IP alebo názvom hostiteľa.

-M Cieľová maska podsiete.

-g Kontrola smerovania zdroja: y alebo n.

-c Protokol. Hodnoty môžu byť udp, icmp, tcp, tcp/ack, ospf, pip, esp, ah a all.

-o Zdrojový port alebo operácia typu ICMP.

-p Zdrojový port alebo hodnota typu ICMP.

-O Cieľový port alebo operácia kódu ICMP.

-P Cieľový port alebo hodnota kódu ICMP.

-r Smerovanie:

- **r** Pakety poslané ďalej.
- **l** Pakety lokálneho určenia/pôvodu.
- **b** Oboje.

-l Riadenie protokolu.

- **y** Zahrnúť do protokolu.
- **n** Nepridávať do protokolu

-f Fragmentácia.

- **y** Použije sa pre hlavičky fragmentov, fragmenty a nefragmenty.
- **o** Použije sa pre fragmenty a hlavičky fragmentov.
- **n** Použije sa iba pre nefragmenty.
- **h** Použije sa iba pre nefragmenty a hlavičky fragmentov.

-t ID tunela.

-i Rozhranie, ako je napríklad tr0 alebo en0.

Ďalšie informácie nájdete v popis príkazov **genfilt** a **chfilt**.

Automaticky generované pravidlá filtrovania a užívateľom zadané pravidlá filtrovania:

Niektoré pravidlá sa generujú automaticky na použitie filtra bezpečnosti IP a kódu tunela.

Automaticky generované pravidlá zahŕňajú nasledujúce množiny pravidiel:

- Pravidlá pre démona kľúčov relácií, ktorý obnovuje kľúče IPv4 v IKE
- Pravidlá pre spracovanie paketov AH a ESP.

Pravidlá filtrovania sa generujú tiež automaticky pri definovaní pravidiel. Pri manuálnych tuneloch určujú automaticky generované pravidlá zdrojové a cieľové adresy a hodnoty masky, rovnako ako ID tunela. Všetky prenosy medzi týmito adresami prechádzajú cez tunel.

Pri tuneloch IKE určujú automaticky generované pravidlá filtrovania protokol a čísla portov počas vyjednávania IKE. Pravidlá filtrovania IKE sú uložené v oddelenej tabuľke, ktorá sa prehľadáva po statických pravidlách filtrovania a pred automaticky generovanými pravidlami. Pravidlá filtrovania IKE sa vkladajú do predvolenej pozície v rámci statickej tabuľky filtrovania, no užívateľ ich môže premiestniť.

Automaticky generované pravidlá povoľujú v rámci tunela všetky prenosy. Užívateľom definované pravidlá môžu na niektoré typy prenosov aplikovať obmedzenia. Tieto užívateľom definované pravidlá umiestnite pred automaticky generované pravidlá, pretože funkcia IP Security používa prvé nájdené pravidlo, ktoré sa vzťahuje na príslušný paket. Nižšie je uvedený príklad užívateľom definovaných pravidiel filtrovania, ktoré filtrujú prenosy na základe operácie ICMP.

```
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
```

Za účelom zjednodušenia konfigurácie tunelov sa pravidlá filtrovania automaticky generujú pri definovaní daných tunelov. Túto funkciu možno potlačiť zadaním príznaku **-g** v **gentun**. Pomocou príkazov **genfilt** nájdete súbor vzorového filtra na generovanie filtrovacích pravidiel pre rôzne služby TCP/IP v /usr/samples/ipsec/filter.sample.

Preddefinované pravidlá filtrovania:

Niekoľko preddefinovaných pravidiel filtrovania sa generuje automaticky s určitými udalosťami.

Pri načítaní zariadenia ipsec_v4 alebo ipsec_v6 sa do tabuľky filtrovania vloží preddefinované pravidlo, ktoré sa následne aktivuje. V predvolenom nastavení povoľuje toto preddefinované pravidlo všetky pakety, no užívateľ ho môže nakonfigurovať tak, aby všetky pakety zakázalo.

Poznámka: Pri vzdialenej konfigurácii skontrolujte ešte pred dokončením konfigurácie, či nie je zapnuté pravidlo odmietnutia, aby ste svojej relácii zabránili zamknutie mimo počítača. Vzniku tejto situácie možno zabrániť nastavením možnosti povolenia ako predvolenej akcie alebo vykonaním konfigurácie tunela do vzdialeného počítača pred aktiváciou funkcie IP Security.

Obe tabuľky filtra IP verzie 4 aj 6 majú preddefinované pravidlo. Obidve pravidlá možno nezávisle zmeniť tak, aby zakazovali všetky prenosy. Prenosy budú zablokované, až kým nebudú špecificky definované inými pravidlami filtrovania. Jedinou ďalšou možnosťou na zmenu preddefinovaných pravidiel je **chfilt** s voľbou **-l**, čo umožňuje protokolovanie paketov zhodujúcich sa s týmto pravidlom.

Na podporu tunelových prepojení IKE sa do tabuľky filtrov IP verzie 4 umiestni dynamické pravidlo filtrovania. Predstavuje pozíciu, do ktorej sa umiestňujú dynamické pravidlá filtrovania v tabuľke filtrovania. Užívateľ môže túto pozíciu kontrolovať premiestnením jej polohy nahor alebo nadol v tabuľke filtrovania. Po inicializácii démona správy

tunelového prepojenia a démona **isakmpd** za účelom povolenia vyjednávania tunelových prepojení IKE sa v tabuľke dynamických filtrov automaticky vytvoria pravidlá na prácu so správami IKE a paketmi AH a ESP.

Masky podsiete:

Masky podsiete sa používajú na vytvorenie množiny ID, ktoré sú priradené k pravidlu filtrovania. Hodnota masky je ANDed s ID v pravidlách filtrovania a porovná sa s ID, ktoré je špecifikované v pakete.

Napríklad, pravidlo filtrovania so zdrojovou IP adresou 10.10.10.4 a maskou podsiete 255.255.255.255 špecifikovalo, že musí dôjsť k presnej zhode v desiatkovej IP adrese, ako to vidíte ďalej:

	Binárna hodnota	**Decimálna hodnota**
Zdrojová adresa IP	1010.1010.1010.0100	10.10.10.4
Maska podsiete	11111111.11111111.11111111.11111111	255.255.255.255

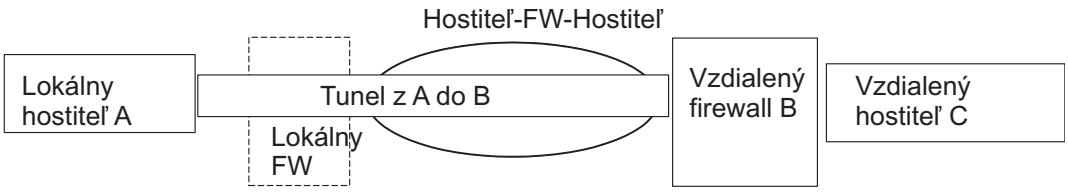
Podsieť 10.10.10.x je špecifikovaná ako 11111111.11111111.11111111.0 alebo 255.255.255.0. Prichádzajúca adresa by mala aplikovanú masku podsiete, kombinácia by sa následne porovnala s ID v pravidle filtrovania. Napríklad, z adresy 10.10.10.100 vznikne po aplikovaní masky podsiete adresa 10.10.10.0, ktorá zodpovedá pravidlu filtrovania.

Maska podsiete 255.255.255.240 môže na mieste posledných štyroch bitov v adrese obsahovať akúkoľvek hodnotu.

Konfigurácia hostiteľ-firewall-hostiteľ:

Voľba konfigurácie hostiteľ-firewall-hostiteľ pre tunely umožňuje vytvoriť tunel medzi hostiteľom a bránou firewall a následne automaticky vygenerovať potrebné pravidlá filtrovania pre správnu komunikáciu medzi vašim hostiteľom a hostiteľom za bránou firewall.

Automaticky generované pravidlá filtrovania povoľujú v zadanom tuneli všetky pravidlá medzi dvoma hostiteľmi bez brány firewall. Predvolené pravidlá pre hlavičky UDP (user datagram protocol), AH (Authentication Headers) a ESP (Encapsulating Security Payload) by mali spracovať hostiteľa pre komunikáciu firewall. Pre dokončenie nastavenia je potrebné vykonať príslušnú konfiguráciu brány firewall. Na zadanie hodnôt SPI a kľúčov potrebných pre firewall by ste mali použiť exportný súbor z tunela, ktorý ste vytvorili.



Obrázok 14. Konfigurácia typu hostiteľ-firewall-hostiteľ

Tento nákres zobrazuje konfiguráciu typu hostiteľ-firewall-hostiteľ. Hostiteľ A obsahuje tunel prevádzkovaný cez lokálnu bránu firewall smerom do siete Internet. Následne prechádza k vzdialenej bráne firewall B a potom k vzdialenému hostiteľovi C.

Protokolovacie zariadenia

Pri vzájomnej komunikácii hostiteľov môžu byť prenášané pakety zaprotokolované do démona systémového protokolu syslogd. Zobrazia sa aj ďalšie dôležité správy o protokoloch IP Security.

Administrátor môže zaprotokolované informácie sledovať, pretože mu môžu slúžiť ako pomôcka na analýzu prenosu a ladenie. Ak chcete nastaviť prostriedky pre protokolovanie, použite nasledovný postup.

1. Upravte súbor /etc/syslog.conf pridaním nasledovnej položky:

```
local4.debug var/adm/ipsec.log
```

Na zaznamenanie prenosov a udalostí protokolov IP Security použite prostriedok local4. Používajú sa štandardné úrovne priorít operačného systému. Ak je prenos cez tunelové prepojenia a filtre protokolov IP Security stabilný a správny, mali by ste použiť úroveň priority debug.

Poznámka: Protokolovanie udalostí filtra môže vytvoriť dôležitú aktivitu na hostiteľovi IP Security a spotrebovať veľký rozsah pamäte.

2. Uložte /etc/syslog.conf file.
3. Prejdite do adresára zadaného pre protokolový súbor a vytvorte prázdny súbor s rovnakým názvom. Vo vyššie uvedenom prípade by ste prešli do adresára /var/adm a zadali príkaz:

   ```
   touch ipsec.log
   ```

4. Vydajte príkaz **refresh** pre podsystém syslogd:

   ```
   refresh -s syslogd
   ```

5. Ak používate tunely IKE, skontrolujte, či súbor /etc/isakmpd.conf uvádza želanú úroveň protokolovania **isakmpd**. (Bližšie informácie o protokolovaní IKE nájdete v “Diagnostika problémov s bezpečnosťou internetového protokolu” na strane 250.)
6. Ak by ste pri vytváraní filtrovacích pravidiel pre svojho hostiteľa chceli zaprotokolovať pakety zhodujúce sa s určitým pravidlom, nastavte parameter *-l* pre pravidlo na **Y** (Yes) pomocou príkazov **genfilt** alebo **chfilt**.
7. Zapnite protokolovanie paketu a spustite démona **ipsec_logd** pomocou príkazu:

   ```
   mkfilt -g start
   ```

   Protokolovanie paketu môžete zastaviť vydaním tohto príkazu:

   ```
   mkfilt -g stop
   ```

Nasledovný vzorový protokolový súbor obsahuje záznamy o prenose a ďalšie záznamy protokolu IP Security:

```
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
```

Záznamy protokolu sú vysvetlené v nasledujúcich odsekoch.

1 Aktivácia démona protokolovania filtrov.

2 Zapnutie protokolovania paketov filtra pomocou príkazu **mkfilt -g start**.

3 Aktivácia tunelového prepojenia, zobrazuje sa ID tunelového prepojenia, zdrojová adresa, cieľová adresa a časová známka.

4-9 Aktivácia filtrov. Protokolovanie zobrazuje všetky zavedené pravidlá filtrovania.

10 Správa o aktivácii filtrov.

11-12 Tieto záznamy zobrazujú vyhľadávací kód DNS pre hostiteľa.

13-15 Tieto záznamy sčasti zobrazujú pripojenie Telnet (ostatné záznamy boli v tomto príklade z priestorových dôvodov odstránené).

16-19 Tieto záznamy zobrazujú dve operácie ping.

20 Zavretie démona protokolovania filtrov.

Nasledovný príklad zobrazuje dvoch hostiteľov vyjednávajúcich tunelové prepojenia fázy 1 a fázy 2 z hľadiska iniciátorského hostiteľa. (Úroveň protokolovania **isakmpd** bola zadaná ako **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
```

Záznamy protokolu sú vysvetlené v nasledujúcich odsekoch.

1-2 Príkaz **ike cmd=activate phase=1** iniciuje pripojenie.

3-10 Démon **isakmpd** vyjednáva tunelové prepojenie fázy 1.

11-12 Nástroj Tunnel Manager prijíma platné priradenie zabezpečenia fázy 1 od respondenta.

13 Nástroj Tunnel Manager kontroluje, či pre parameter **ike cmd=activate** existuje hodnota fázy 2 pre ďalšiu činnosť. Hodnota neexistuje.

14-16 Démon **isakmpd** dokončuje vyjednávanie fázy 1.

17-21 Príkaz **ike cmd=activate phase=2** iniciuje tunelové prepojenie fázy 2.

22-29 Démon **isakmpd** vyjednáva tunelové prepojenie fázy 2.

30-31 Nástroj Tunnel Manager prijíma platné priradenie zabezpečenia fázy 2 od respondenta.

32 Nástroj Tunnel Manager zapisuje dynamické pravidlá filtrovania.

33 Príkaz **ike cmd=list** zobrazuje tunelové prepojenia IKE.

Štítky v položkách polí:

Polia v položkách protokolu sú skrátené s cieľom zredukovať priestorové požiadavky DASD.

Pole	Význam
#	Číslo pravidla, ktoré zaistilo zaprotokolovanie tohto paketu.
R	Typ pravidla **p** Povoliť **d** Zakázať
i/o	Smer paketu zachyteného podporným kódom filtra. Určuje adresu IP adaptéra priradeného k paketu: • V prípade prichádzajúcich (i) paketov je to adaptér, na ktorý paket prichádza. • Pre odchádzajúce pakety (o) je toto adaptér, ktorý by mal na základe určenia vrstvy IP spracovať prenos paketu.
s	Určuje adresu IP odosielateľa paketu (extrahovanú z hlavičky IP).
d	Určuje adresu IP určeného príjemcu paketu (extrahovanú z hlavičky IP).
p	Určuje protokol vyššej úrovne slúžiaci na vytvorenie správy v údajovej časti paketu. Možno číslo alebo názov, napríklad: udp, icmp, tcp, tcp/ack, ospf, pip, esp, ah alebo all.
sp/t	Určuje číslo portu protokolu priradené k odosielateľovi paketu (extrahované z hlavičky TCP/UDP). Ak ide o protokol ICMP alebo OSPF, hodnota sa nahradí hodnotou **t**, ktorá určuje typ protokolu IP.
dp/c	Určuje číslo portu protokolu priradené k určenému príjemcovi paketu (extrahované z hlavičky TCP/UDP). Ak ide o protokol ICMP, hodnota sa nahradí hodnotou **c**, ktorá určuje kód IP.
-	Určuje, že k dispozícii nie sú žiadne informácie

Pole	Význam	
r	Naznačuje, či paket má lokálne určenie.	
	f	Postúpené pakety
	l	Lokálne pakety
	o	Odchádzajúce
	b	Oboje
l	Určuje dĺžku príslušného paketu v bajtoch.	
f	Určuje, či paket je fragmentom.	
T	Označuje ID tunelového prepojenia.	
i	Určuje vstupné rozhranie pre paket.	

Protokolovanie Internet Key-Exchange:

Môžete zapnúť protokolovanie udalostí Internet Key-Exchange do zariadenia SYSLOG s démonom **isakmpd**.

Pre démona **isakmpd** zapínate protokolovanie pomocou príkazu **ike cmd=log**. Úroveň protokolovania môžete nastaviť v konfiguračnom súbore /etc/isakmpd.conf s parametrom **log_level**. V závislosti od množstva informácií, ktoré chcete protokolovať, môžete túto úroveň nastaviť na *none*, *errors*, *isakmp_events* alebo *information*.

Ak chcete napríklad uviesť, že chcete protokolovať informácie o protokole a implementácii, zadajte parameter:

```
log_level=INFORMATION
```

Démon **isakmpd** spustí jeden z dvoch procesov: zašle alebo zhodnotí návrh. Ak je návrh prijatý, vytvorí sa priradenie zabezpečenia a nastaví sa tunelové prepojenie. Ak návrh nie je prijatý alebo sa pripojenie skončí pred dokončením vyjednávania, démon **isakmpd** označí chybu. Položky v zariadení SYSLOG z **tmd** určujú, či bolo rokovanie úspešné. Zlyhanie spôsobené neplatným certifikátom sa zaprotokoluje do zariadenia SYSLOG. Ak chcete zistiť presnú príčinu neúspešného rokovania, pozrite si údaje v protokolovacom súbore zadané v /etc/syslog.conf.

Zariadenie SYSLOG pridáva predponu do každého riadku protokolu a registruje dátum, čas, počítač a program. V nasledujúcom príklade sa ako názov počítača použila hodnota googly a ako názov programu sa použila hodnota 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
```

Ak chcete zlepšiť jas, na výber riadkov protokolu, o ktoré máte záujem (napríklad celé protokolovanie **isakmpd**), použite príkaz **grep** a na odstránenie predpony z každého riadku použite príkaz **cut**.

Súbor /etc/isakmpd.conf:

Môžete nakonfigurovať voľby pre démona **isakmpd** v súbore /etc/isakmpd.conf.

V súbore /etc/isakmpd.conf sú k dispozícii tieto voľby.

Konfigurácia protokolu

Určite si rozsah informácií, ktoré chcete protokolovať a nastavte úroveň. Démony IKE používajú túto voľbu na zadanie úrovne protokolovania.

Syntax: none | error | isakmp_events | information

kde level má tento význam:

none Protokolovanie sa nevykonáva. Ide o predvolenú hodnotu.

error Chyby protokolovania protokolu alebo chyby aplikačného programového rozhrania (API).

isakmp_events
Protokolovať udalosti alebo chyby protokolu IKE. Túto úroveň použite pri ladení problému.

information
Protokolovať informácie o protokole a implementácii.

Rokovanie o nerozlíšenej adrese IP
Túto voľbu môžete nastaviť na hodnotu YES alebo NO. Keď nastavíte túto voľbu na hodnotu YES, lokálna databáza IKE musí obsahovať IP adresy pre tunel phase-1 aj pre koncové body. Ak chcete prijať prichádzajúci tunel hlavného režimu, musíte pre hostiteľa zadať YES. Adresou IP môže byť primárny ID alebo voliteľná adresa IP, ktorá je priradená k nejakému inému typu ID.

Ak chcete prijať prichádzajúce pripojenie hlavného režimu, nastavte túto voľbu na NO. Ak nastavíte túto voľbu na NO, hostiteľ môže prijať pripojenie aj vtedy, ak databáza IKE neuvádza adresy IP pre koncové body 1. fázy. Aby však hostiteľ prijal pripojenie, musíte použiť autentifikáciu pomocou certifikátu. Umožňuje to hostiteľovi s dynamicky priradenou adresou IP iniciovať tunel hlavného režimu do počítača.

Ak tento parameter neuvádzate, predvolená hodnota je NO.

Syntax: MAIN_MODE_REQUIRES_IP= YES | NO

Konfigurácia servera SOCKS4
Voľba SOCKS4_PORTNUM je voliteľná. Ak ju neuvediete, použije sa predvolená hodnota portu servera SOCKS 1080. Hodnota portu sa použije, keď server SOCKS komunikuje so serverom HTTP.

Syntax: *mnemonic* = *value*
kde *mneumonic* a *value* môžu mať tieto hodnoty:

SOCKS4_SERVER= uvádza názov servera

SOCKS4_PORTNUM= uvádza číslo portu servera SOCKS

ID užívateľa SOCKS4_USERID=

Konfigurácia servera LDAP

Syntax: *mnemonic* = *value*
kde *mnemonic* a *value* môžu mať tieto hodnoty:

LDAP_SERVER= uvádza názov servera LDAP

LDAP_VERSION= verzia servera LDAP (môže byť 2 alebo 3)

LDAP_SERVERPORT= číslo portu servera LDAP

Hodnota uplynutia času vyhradeného na hľadanie klienta LDAP_SEARCHTIME=

Poradie výberu CRL
Ak sú nakonfigurované oba servery, táto voľba definuje, či sa najprv bude dotazovať server HTTP alebo LDAP. Voľba CRL_FETCH_ORDER je voliteľná. Predvoleným poradím výberu v prípade, že sú nakonfigurované oba servery, je najprv výber servera HTTP a potom výber servera LDAP.

Syntax: CRL_FETCH_ORDER= *protocol#*, *protocol#*
kde *protocol#* môže byť HTTP alebo LDAP.

Špecifikácia portu IKEv1 a IKEv2
Tento reťazec zadáva porty používané démonom **isakmpd** (IKEv1) a démonom **ikev2d** (IKEv2). Démon **iked** (démon brokera správ IKE) vyhľadá túto položku a spustí démonov **isakmpd** a **ikev2d** na príslušných portoch.

Syntax: v1=port-natport,v2=port-natport

Diagnostika problémov s bezpečnosťou internetového protokolu
Nasledujú rady a tipy, ktoré by mohli pomôcť, keď narazíte na problém.

Nastavenie protokolovania pri prvej konfigurácii IPSec. Protokoly sú mimoriadne užitočné pri určovaní problémov s filtrami a tunelovými prepojeniami. (Podrobné informácie o protokoloch nájdete v časti “Protokolovacie zariadenia” na strane 245.)

Ak chcete určiť, ktoré démony bezpečnosti IP sú spustené, zadajte tento príkaz:

```
ps -ef
```

K bezpečnosti IP sú priradené tieto démony: **tmd**, **iked**, **isakmpd**, **ikev2d**, **cpsd**.

Poznámka: Ak sú IKEv1 aj IKEv2 nakonfigurované, démon **iked** je spustený. V opačnom prípade je spustený démon **iskmpd** alebo démon **ikev2d**. Táto konfigurácia je v súbore **/etc/isakmpd.conf**.

Riešenie problémov pri chybách manuálneho tunela:

Nasledujú opisy niekoľkých možných chýb tunelových prepojení spolu s ich riešeniami.

Chyba	Možný problém a riešenie
V dôsledku zadania príkazu **mktun** sa vyskytla nasledovná chyba: insert_tun_man4(): write failed : The requested resource is busy.	Problém: Tunelové prepojenie, ktoré ste chceli aktivovať, je už aktívne, alebo hodnoty indexu SPI kolidujú. Oprava: Pomocou príkazu **rmtun** deaktivujte a potom pomocou príkazu mktun znovu aktivujte tunelové prepojenie. Skontrolujte, či sa hodnoty indexu SPI pre tunelové prepojenie nezhodujú s hodnotami pre iné tunelové prepojenie. Každé tunelové prepojenie by malo mať jedinečné hodnoty indexu SPI.

V dôsledku zadania príkazu **mktun** sa vyskytla nasledovná chyba:
 	
Device ipsec_v4 is in Defined status.
 	
Tunnel activation for IP Version 4 not performed.
	Problém: Nesprístupnili ste zariadenie zabezpečenia IP.
	
	Oprava: Zadajte nasledovný príkaz:
	`mkdev -l ipsec -t 4`
	
	Možno budete musieť zmeniť voľbu **-t** na 6, ak sa prejaví tá istá chyba pri aktivácii tunela IP verzie 6. Zariadenia musia byť dostupné. Ak chcete skontrolovať stav zariadenia IP Security, zadajte nasledovný príkaz:
	`lsdev -Cc ipsec`

V dôsledku zadania príkazu **gentun** sa vyskytla nasledovná chyba: Invalid Source IP address	Problém: Nezadali ste platnú zdrojovú adresu IP. Oprava: Ak ide o tunelové prepojenie IPv4, skontrolujte, či ste zadali dostupnú adresu IPv4 pre lokálny počítač. Pri generovaní tunelov nemôžete použiť názvy hostiteľa pre zdroj; môžete použiť iba názvy hostiteľa pre cieľ. Ak ide o tunelové prepojenie IPv6, skontrolujte, či ste zadali dostupnú adresu IPv6. Ak napíšete netstat -in a neexistujú žiadne adresy IP verzie 6, spustite /usr/sbin/autoconf6 (rozhranie) pre automaticky generovanú adresu lokálnej linky (s použitím adresy MAC), alebo použite príkaz **ifconfig** pre manuálne priradenie adresy.
V dôsledku zadania príkazu **gentun** sa vyskytla nasledovná chyba: Invalid Source IP address	Problém: Nezadali ste platnú zdrojovú adresu IP. Oprava: Ak ide o tunelové prepojenie IPv4, skontrolujte, či ste zadali dostupnú adresu IPv4 pre lokálny počítač. Pri generovaní tunelových prepojení môžete názvy hostiteľov používať len pre cieľ a nemôžete ich používať pre zdroj. Ak ide o tunelové prepojenie IPv6, skontrolujte, či ste zadali dostupnú adresu IPv6. Ak napíšete netstat -in a neexistujú žiadne adresy IP verzie 6, spustite /usr/sbin/autoconf6 (rozhranie) pre automaticky generovanú adresu lokálnej linky (s použitím adresy MAC), alebo použite **ifconfig** pre manuálne priradenie adresy.
V dôsledku zadania príkazu **mktun** sa vyskytla nasledovná chyba: insert_tun_man4(): write failed : A system call received a parameter that is not valid.	Problém: Tunelové prepojenie sa generovalo s použitím neplatnej kombinácie ESP a AH alebo bez potrebného použitia nového formátu hlavičky. Oprava: Overte si, ktoré algoritmy autentifikácie používa príslušný tunel. Nezabúdajte, že algoritmy HMAC_MD5 a HMAC_SHA vyžadujú nový formát hlavičky. Nový formát hlavičky možno zmeniť pomocou rýchlej cesty SMIT **ips4_basic** alebo pomocou parametra *-z* s príkazom **chtun**. Pamätajte tiež na to, že DES_CBC_4 nemôže byť použité s novým formátom hlavičky.
V dôsledku pokusu o použitie technológie IP Security sa vyskytla nasledovná chyba: The installed bos.crypto is back level and must be updated.	Problém: Súbory bos.net.ipsec.* boli aktualizované na novšiu verziu, ale zodpovedajúce súbory bos.crypto.* neboli aktualizované. Oprava: Aktualizujte súbory bos.crypto.* na verziu zodpovedajúcu aktualizovaným súborom bos.net.ipsec.*.

Riešenie problémov pri chybách tunela IKE (Internet Key Exchange):

Nasledujúce časti opisujú chyby, ktoré sa môžu vyskytnúť pri používaní tunelov IKE (Internet Key Exchange).

Tok spracovania tunela Internet Key Exchange:

Táto časť opisuje tok procesu pre tunel Internet Key Exchange.

Tunely IKE sa nastavujú prostredníctvom komunikácie príkazu **ike** s nasledujúcimi démonmi:

tmd démon nástroja Tunnel Manager

iked Démon brokera IKE (aktívny len keď je na systéme nakonfigurovaný démon IKEv1 aj IKEv2)

isakmpd
Démon IKEv1

ikev2d Démon IKEv2

cpsd démon servera proxy pre certifikáty

Aby boli tunely IKE správne nastavené, musia byť spustené démony **tmd** a **isakmpd**. Ak je technológia IP Security nastavená na spúšťanie pri opakovanom spustení, tieto démony sa spustia automaticky. V opačnom prípade sa musia spustiť zadaním nasledujúceho príkazu:

```
startsrc -g ike
```

Nástroj Tunnel Manager zadá démonu **isakmpd** požiadavky na spustenie tunelového prepojenia. Ak tunelové prepojenie už existuje, alebo ak nie je platné (napríklad vzdialená adresa je neplatná), oznámi sa chyba. Ak sa začalo vyjednávanie, môže sa dokončiť až po uplynutí určitého času, ktorý závisí od reakčnej doby siete. Na zistenie úspešnosti vyjednávania je možné použiť príkaz **ike cmd=list**, ktorý zobrazí stav tunelového prepojenia. Aj manažér tunela protokoluje udalosti syslog do úrovní debug, event a information, ktoré možno použiť na monitorovanie postupu vyjednávania.

Postup je nasledovný:

1. Zadaním príkazu **ike** inicializujte tunel.
2. Démon **tmd** zadá pre démon **isakmpd** požiadavku na pripojenie pre správu kľúčov (fáza 1).
3. Démon **isakmpd** odpovedá s SA created alebo chybovým hlásením.
4. Démon **tmd** zadá pre démon **isakmpd** požiadavku na pripojenie pre tunelové prepojenie na správu údajov (fáza 2).
5. Démon **isakmpd** odpovedá s SA created alebo chybovým hlásením.
6. Parametre tunelového prepojenia sa vložia do vyrovnávacej pamäte jadra pre tunelové prepojenie.
7. Pravidlá filtra sa pridajú do dynamickej tabuľky jadra pre filtre.

Ak počítač vystupuje ako respondent, démon **isakmpd** upozorní démon nástroja Tunnel Manager **tmd** na úspešné vyjednanie tunelového prepojenia a vloženie nového tunelového prepojenia do jadra. V tom prípade sa proces začína od kroku 3 a pokračuje až po krok 7 (bez zadávania požiadaviek démona **tmd** na pripojenie).

Funkcia protokolovania Parse Payload:

Priradenie zabezpečenia medzi dvoma koncovými bodmi sa vytvorí výmenou správ IKE. Funkcia Parse Payload analyzuje správy vo formáte čitateľnom pre ľudí.

Protokolovanie Parse Payload možno zapnúť úpravou súboru /etc/isakmpd.conf. Záznam v protokolovom súbore /etc/isakmpd.conf vyzerá približne nasledovne:

```
information
```

Typ údajovej časti IKE zaznamenávaný funkciou Parse Payload závisí od obsahu správy IKE. Príklady obsahujú údajovú časť pre priradenie zabezpečenia, výmenu kľúčov, žiadosť o certifikát, certifikát a pre podpis. Na tomto mieste je uvedený príklad s protokolom funkcie Parse Payload, v ktorom sa za hlavičkou ISAKMP_MSG_HEADER nachádza päť údajových častí:

```
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)
```

V každej hodnote payload, pole **Next Payload** ukazuje na hodnotu payload nasledujúcu za aktuálnou hodnotou. Ak ide o poslednú údajovú časť v správe IKE, v poli **Next Payload** sa nachádza nulová hodnota (None).

Každá údajová časť v príklade obsahuje informácie týkajúce sa prebiehajúcich vyjednávaní. Napríklad SA Payload má Proposal a Transform Payloads, ktoré zas zobrazujú šifrovací algoritmus, režim autentifikácie, hašovací algoritmus, typ životnosti SA a trvanie SA, ktoré iniciátor navrhuje respondentovi.

Údajová časť priradenia zabezpečenia pozostáva z jednej alebo viacerých údajových častí pre návrh a transformáciu. Hodnota v poli **Next Payload** pre údajovú časť pre návrh je 0, ak ide o jedinú údajovú časť pre návrh, alebo 2, ak ide o viacero údajových častí pre návrh. Podobne aj hodnota v poli **Next Payload** pre údajovú časť pre transformáciu je 0, ak ide o jedinú údajovú časť pre transformáciu, alebo 3, ak ide o viacero údajových častí pre transformáciu, ako je uvedené aj v nasledujúcom príklade:

```
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)
```

Hlavička správy IKE protokolu Parse Payload zobrazuje typ výmeny (hlavný režim a agresívny režim), dĺžku celej správy, identifikátor správy atď.

Údajová časť žiadosti o certifikát vyžaduje certifikát od respondenta. Respondent odosiela certifikát v osobitnej správe. V nasledujúcom príklade sú uvedené údajové časti pre certifikát a podpis, ktoré sa odosielajú partnerovi v rámci vyjednávania priradenia zabezpečenia. Údaje o certifikáte a podpise sú vytlačené v hexadecimálnom formáte.

```
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
```

Problémy s digitálnymi certifikátmi a podpisovým režimom:

Nasledujúce informácie popisujú riešenia možných problémov s digitálnymi certifikátmi a režimom podpisovania, s ktorými sa môžete stretnúť:

Chyba	Možný problém a riešenie
Chyba: Nespustí sa **cpsd** (démon Certificate Proxy Server). V protokolovom súbore sa zobrazí záznam podobný nasledovnému: Sep 21 6:02:00 ripple CPS[19950]: Init():Lo adCaCerts() failed, rc =-12	Problém: Databáza certifikátov sa neotvorila, alebo sa nenašla. Oprava: Uistite sa, že databázy certifikátov nástroja Key Manager sa nachádzajú na ceste /etc/security. Databáza pozostáva z nasledovných súborov: ikekey.crl, ikekey.kdb, ikekey.rdb a ikekey.sth. Ak chýba len súbor ikekey.sth, voľba stash password nebola vybraná, keď bola vytvorená databáza nástroja Key Manager. Ak chcete umožniť používanie digitálnych certifikátov s technológiou IP Security, heslo musí byť skryté. (Ďalšie informácie nájdete v časti Vytvorenie databázy kľúčov.)
Chyba: Nástroj Key Manager vydá nasledujúcu chybu pri prijímaní certifikátu: Invalid Base64-encoded data was found	Problém: v súbore certifikátu boli zistené nadbytočné údaje alebo sa ďalšie údaje stratili alebo boli poškodené. Oprava: Certifikát s kódovaním 'DER' musí byť ohraničený nasledovnými reťazcami (pozrite nižšie). Pred textom ohraničeným reťazcami BEGIN CERTIFICATE a END CERTIFICATE a za ním sa nesmú nachádzať žiadne znaky. -----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----- Pri diagnostikovaní a odstraňovaní tohto problému máte vám môžu pomôcť nasledovné možnosti. • Ak sú údaje stratené alebo poškodené, znovu vytvorte certifikát. • Úspešnou analýzou certifikátu pomocou analyzátora ASN.1 (dostupného na stránkach www) skontrolujte platnosť certifikátu.
Chyba: Nástroj Key Manager vydá nasledujúcu chybu pri prijímaní osobného certifikátu: No request key was found for the certificate	Problém: Pre prijatý osobný certifikát neexistuje príslušná žiadosť. Oprava: Znovu vytvorte žiadosť o osobný certifikát a požiadajte o nový certifikát.
Chyba: Zlyhá vyjednávanie IKE a v protokolovom súbore sa objaví položka podobná tejto: inet_cert_service:: channelOpen(): clientInitIPC():error,rc =2 (No such file or directory)	Problém: **cpsd** nie je spustený alebo bol zastavený. Oprava: Spustite IP Security, čím sa spustia príslušní démoni.
Chyba: Zlyhá vyjednávanie IKE a v protokolovom súbore sa objaví položka podobná tejto: CertRepo::GetCertObj: DN Does Not Match: ("/C=US/O=IBM/CN=ripple.austin.ibm.com")	Problém: Charakteristický názov X.500 zadaný počas definovania tunelového prepojenia IKE sa nezhoduje s hodnotou v osobnom certifikáte. Oprava: Zmeňte definíciu tunela IKE tak, aby sa zhodovala s jedinečným názvom v certifikáte.

Funkcie sledovania:

Sledovanie je funkcia ladenia pre sledovanie udalostí jadra. Sledovania sa môžu použiť na získanie podrobnejších údajov o udalostiach alebo chybách v kóde filtrov a tunelových prepojení v jadre.

Prostriedok na sledovanie protokolov IP SMIT je k dispozícii v ponuke Advanced IP Security Configuration. Informácie zaznamenané týmto prostriedkom sledovania zahŕňa informácie o chybách, filtroch, filtrovacích informáciách, tuneloch, tunelovacích informáciách, kapsulácii a dekapsulácii, kapsulačných informáciách, šifrovaní a

šifrovacích informáciách. Sledovanie chýb poskytuje tie najdôležitejšie informácie. Sledovanie informácií môže generovať kľúčové informácie a môže mať vplyv na výkon systému. Toto sledovanie poskytuje informácie o zdroji problému a vyžaduje sa aj pri vysvetľovaní problému servisnému technikovi.

Ak chcete povoliť sledovanie, nakonfigurujte zariadenia IPSec a úroveň sledovania všetkých podkomponentov IPSec nastavte na úroveň 7, aby sa generovali užitočné údaje o sledovaní jadra. Ak zariadenia IPSec nie sú nakonfigurované, príkaz na riadenie sledovania komponentov nebude uvádzať záznamy súvisiace so zariadeniami IPSec. Na spustenie sledovania IPSec môžete použiť rýchlu cestu **smit ips4_start** (pre protokol IP verzia 4) alebo **smit ips6_start** (pre protokol IP 6) rozhrania SMIT.

Poznámka: Ak sledovanie komponentov IPSec nie je správne nastavené, zaznamenané sledovania budú prázdne.

Ak chcete, aby sa zaznamenávali údaje o sledovaní jadra, postupujte takto:

1. Vyhľadajte všetky komponenty a pozrite si aktuálne nastavenia úrovne sledovania:
   ```
   # ctctrl -q
   ```
2. Skontrolujte komponent IPSec a jeho podkomponenty. Komponenty budú najskôr uvedené nasledujúcim spôsobom, pričom predvolená úroveň sledovania je 3. Ak chcete zobraziť úvodnú predvolenú úroveň sledovania komponentov, zadajte:
   ```
   # ctctrl -q -c ipsec -r
   ```

Component Name	Have Alias	Memory Trace/Level	System Track/Level	Buffer Size/Allocated
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. Zvýšte úroveň sledovania komponentu IPSec a jeho podkomponentov na úroveň 7, aby bolo podporované sledovanie jadra, zadaním:
   ```
   # ctctrl systracelevel=7 -c ipsec -r
   ```
4. Ak sa chcete utvrdiť v tom, že bola zmenená úroveň sledovania komponentu IPSec a jeho podkomponentov, zadajte:
   ```
   # ctctrl -q -c ipsec -r
   ```

Component Name	Have Alias	Memory Trace/Level	System Track/Level	Buffer Size/Allocated
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

Prístup k prostriedku na sledovanie získate prostredníctvom rýchlej cesty k nástroju SMIT **smit ips4_tracing** (pre verziu IPv4) alebo **smit ips6_tracing** (pre verziu IPv6). Sledovania jadra vytvorené prostredníctvom rýchlej cesty **smit ips4_tracing** alebo **smit ips6_tracing** alebo prostredníctvom prostriedku sledovania z príkazového jadra vygenerujú platné údaje o sledovaní IPSec.

príkaz ipsecstat:

Na vypísanie stavu zariadení IP Security, šifrovacích algoritmov IP Security a štatistiky paketov IP Security môžete použiť príkaz **ipsecstat**.

Vydanie príkazu **ipsecstat** vygeneruje túto vzorovú správu, ktorá zobrazuje, že zariadenia IP Security sú v dostupnom stave, že sú nainštalované tri algoritmy autentifikácie, tri algoritmy šifrovania a vydaná bola aktuálna správa o činnosti paketu. Tieto informácie môžu byť pre vás užitočné, ak pri odstraňovaní problémov s prevádzkou IP Security zisťujete, kde sa nachádza problém.

```
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
```

Poznámka: Už nie je potrebné používať CDMF, pretože algoritmus DES je už dostupný po celom svete. Prekonfigurujte všetky tunelové prepojenia používajúce šifrovanie CDMF tak, aby používali šifrovanie DES alebo Triple DES.

Odkaz na IP Security

Existujú príkazy a metódy pre bezpečnosť IP. Môžete tiež migrovať tunelové prepojenia IKE, filtre a predzdieľané kľúče.

Zoznam príkazov:

Nasledujúca tabuľka poskytuje zoznam príkazov.

Príkaz	Účel
ike cmd=activate	Spustí vyjednávanie Internet Key Exchange (IKE).
ike cmd=remove	Deaktivuje tunely IKE
ike cmd=list	Zobrazí zoznam tunelov IKE
ikedb	Poskytuje rozhranie pre databázu tunelov IKE
gentun	Vytvára definíciu tunelového prepojenia
mktun	Aktivuje definície tunelových prepojení
chtun	Mení definíciu tunelového prepojenia
rmtun	Odstraňuje definíciu tunelového prepojenia
lstun	Zobrazuje zoznam definícií tunelových prepojení
exptun	Exportuje definície tunelových prepojení
imptun	Importuje definície tunelových prepojení
genfilt	Vytvára definíciu filtra
mkfilt	Aktivuje definície filtrov
mvfilt	Presúva pravidlo filtra
chfilt	Mení definíciu filtra

Príkaz	Účel
rmfilt	Odstraňuje definíciu filtra
lsfilt	Zobrazuje zoznam definícií filtrov
expfilt	Exportuje definície filtrov
impfilt	Importuje definície filtrov
ipsec_convert	Zobrazuje stav protokolov IP Security
ipsecstat	Zobrazuje stav protokolov IP Security
ipsectrcbuf	Zobrazuje obsah vyrovnávacej pamäte protokolov IP Security pre sledovanie
unloadipsec	Zruší načítanie šifrovacieho modulu

Zoznam metód:

Nasledujúce informácie uvádzajú zoznam metód.

defipsec
Definuje inštanciu protokolov IP Security pre protokol IPv4 alebo IPv6

cfgipsec
Konfiguruje a načíta parameter **ipsec_v4** alebo **ipsec_v6**

ucfgipsec
Zruší konfiguráciu parametrov **ipsec_v4** alebo **ipsec_v6**

Migrácia zabezpečenia IP:

Tunely IKE, filtre a predzdieľané kľúče môžete migrovať z predchádzajúcich verzií operačného systému AIX.

Migrácia tunelov IKE:

Ak chcete vykonať migráciu tunelov, vykonajte nasledujúce kroky:

1. Spustite skript bos.net.ipsec.keymgt.pre_rm.sh. Keď spustíte tento skript, v adresári /tmp sa vytvoria nasledujúce súbory:
   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

   Upozornenie: Tento skript spustite len raz. Ak zaktualizujete databázu a spustíte tento skript znova, stratíte všetky súbory a nebudete ich môcť znova načítať. Skript v “Skript bos.net.ipsec.keymgt.pre_rm.sh” na strane 260 si prečítajte pred migráciou svojich tunelov.
2. Uložte súbory vytvorené týmto skriptom a súbor /tmp/lpplevel na niektoré externé médium, napríklad na CD alebo disketu.

Migrácia predzdieľaných kľúčov:

Pri aktualizácii formátu predzdieľaných kľúčov postupujte nasledovne.

Databáza predzdieľaného kľúča tunela IKE sa počas migrácie tiež poškodí. Ak chcete zmeniť formát predzdieľaného kľúča, v migrovanom systéme vykonajte nasledujúce kroky:

1. Uložte výstup príkazu **ikedb -g** spustením nasledujúceho príkazu:
   ```
   ikedb -g > out.keys
   ```
2. Upravte súbor out.keys tak, aby FORMAT=ASCII bolo nahradené FORMAT=HEX pre formát predzdieľaného kľúča.

3. Vložte súbor XML spustením nasledovného príkazu:

```
ikedb -pF out.keys
```

Migrácia filtrov:

Vykonajte nasledujúce kroky na migráciu filtrov.

1. Pomocou SMIT vykonaním nasledujúcich krokov vyexportujte súbory s pravidlami filtrovania do adresára /tmp.
   a. Spustite príkaz **smitty ipsec4**.
   b. Vyberte si Advanced IP Security Configuration—>Configure IP Security Filter Rules—>Export IP Security filter rules.
   c. Pre názov adresára zadajte /tmp.
   d. Pod voľbou Filter Rules stlačte F4 a vyberte si zo zoznamu **all**.
   e. Stlačením klávesu Enter uložte pravidlá filtrovania do súboru /tmp/ipsec_fltr_rule.exp na externom médiu.

   Tento postup vykonajte pre všetky systémy, ktoré migrujete z predchádzajúcich verzií operačného systému AIX.
2. Skopírujte šesť súborov tunela vytvorených skriptom, súbor /tmp/lpplevel a súbor /tmp/ipsec_fltr_rule.exp do adresára /tmp na migrovanom systéme.
3. Spustením skriptu bos.net.ipsec.keymgt.post_i.sh znova zaplňte databázu konfiguráciami tunela.
4. Spustením príkazu **ikedb -g** skontrolujte, či sa tunely nachádzajú v databáze.

   Poznámka: Ak informácie o tuneli v databáze nenájdete, spustite znova skript, ale premenujte všetky súbory *.loaded v adresári /tmp na ich pôvodné názvy.

Na migrovanom systéme je po migrácii poškodená databáza filtrov. Ak spustíte na migrovanom systéme príkaz **lsfilt**, dostanete túto chybu:

```
Nemožno získať predvolené pravidlo filtrovania ipv4
```

Ak chcete aktualizovať databázu filtrov, vykonajte tieto kroky:

1. Nahraďte súbory ipsec_filter a ipsec_filter.vc v adresári /etc/security nepoškodenými súbormi z novo migrovaného systému. Ak tieto súbory nemáte, môžete o ne požiadať servis IBM.
2. Pomocou SMIT vykonaním nasledujúcich krokov naimportujte súbory s pravidlami filtrovania do adresára /tmp.
   a. Spustite príkaz **smitty ipsec4**.
   b. Vyberte si Advanced IP Security Configuration—>Configure IP Security Filter Rules—>Import IP Security filter rules.
   c. Pre názov adresára zadajte /tmp.
   d. Pod voľbou **Filter Rules** stlačte **F4** a vyberte si zo zoznamu **all**.
   e. Stlačením klávesu Enter znova vytvorte pravidlá filtrovania. Pravidlá filtrovania môžete vypísať prostredníctvom SMIT alebo pomocou príkazu **lsfilt**.

Skript bos.net.ipsec.keymgt.pre_rm.sh:

Skript bos.net.ipsec.keymgt.pre_rm.sh uloží obsah databázy tunela na systém s operačným systémom AIX.

```
#!/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


#  Copy the database to a save directory in case changes fail
if [ -d /etc/ipsec/inet/DB ]
then
```

```
  cp -R /etc/ipsec/inet/DB /etc/ipsec/inet/DB.sav || exit $?
fi

#  Remember the level you are migrating from
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

#  See if ikedb exists.
if [ -f $IKEDB ]
then

  # If either of the ikedb calls below fails, that's OK.  Just remove the
  # resulting file (which may contain garbage) and continue.  The post_i
  # script will simply not import the file if it doesn't exist, which will
  # mean part or all of the IKE database is lost, but this is preferable
  # to exiting the script with an error code, which causes the entire
  # migration to fail.

  $IKEDB -g > $XMLFILE
  if [ $? -ne 0 ]
  then
    rm -f $XMLFILE  || exit $?
  fi

  if [[ $VR = "5.1" ]]; then
    # This is a special case.  The 5.1 version of ikedb is the only
    # one that does not include preshared keys in the full database
    # output.  So we have to retrieve those separately.
    $IKEDB -g -t IKEPresharedKey > $PSKXMLFILE
    if [ $? -ne 0 ]
    then
     rm -f $PSKXMLFILE  || exit $?
    fi
  fi

#  Make sure ikegui command is installed
elif [ -f /usr/sbin/ikegui ]
then

  #  Get database information and save to /tmp
  /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
```

Skript bos.net.ipsec.keymgt.post_i.sh:

Skript bos.net.ipsec.keymgt.post_i.sh zavedie obsah databázy tunela na migrovaný systém s operačným systémom AIX.

```
#!/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
```

Zabezpečenie NFS (Network File System)

NFS (Network File System) je bežne dostupná technológia, ktorá umožňuje zdieľanie údajov v sieti medzi rôznymi hostiteľmi.

NFS taktiež podporuje použitie autentifikácie Kerberos 5 okrem DES. Bezpečnosť Kerberos 5 sa poskytuje pod mechanizmom protokolu s názvom RPCSEC_GSS.

Okrem štandardných autentifikačných systémov UNIX poskytuje NFS prostriedky na autentifikáciu užívateľov a počítačov v sieťach na báze správa za správou. Tento systém autentifikácie používa šifrovací štandard DES (Data Encryption Standard) a kryptografické verejné kľúče.

NFS taktiež podporuje použitie autentifikácie Kerberos 5 okrem DES. Bezpečnosť Kerberos 5 sa poskytuje pod mechanizmom protokolu s názvom RPCSEC_GSS. Opis a informácie o správe a používaní autentifikácie Kerberos s NFS nájdete v príručke *NFS Administration Guide*.

Všeobecné predpisy pre zabezpečenie systému NFS (Network File System)

Existuje niekoľko pravidiel na pomoc pri zabezpečovaní Network File System (NFS).

- Skontrolujte, či sú nainštalované najnovšie softvérové opravy. Opravy bezpečnostných problémov by mali byť považované za mimoriadne dôležité. Mal by byť udržiavaný celý softvér v danej infraštruktúre. Napríklad nainštalovanie opráv na operačný systém, ale nenainštalovanie opráv na webový server môže poskytnúť útočníkovi spôsob na pripojenie vášho prostredia, čomu by sa dalo vyhnúť, ak by bol aktualizovaný aj webový server. Ak chcete odoberať bezpečnostné výstrahy služby IBM System p a získavať tak najnovšie informácie o bezpečnosti, navštívte nasledujúcu webovú adresu: http://www14.software.ibm.com/webapp/set2/subscriptions/pqvcmjd.
- Nakonfigurujte si server NFS na export systémov súborov s najmenším rozsahom potrebných privilégií. Ak chcú užívatelia len čítať zo systému súborov, nemali by mať možnosť zapisovať doň. Tým možno zmierniť pokus o prepísanie dôležitých údajov, modifikáciu konfiguračných súborov alebo zapísanie svojvoľného spustiteľného kódu do vyexportovaného systému súborov. Zadajte privilégiá pomocou SMIT alebo priamo úpravou súboru /etc/exports.
- Nakonfigurujte server NFS na export systémov súborov explicitne pre užívateľov, ktorí majú mať doň prístup. Väčšina implementácií NFS umožňuje uviesť, ktorí klienti NFS majú mať prístup do daného systému súborov. Tým sa zmiernia pokusy neautorizovaných užívateľov o prístup k systémom súborov. Predovšetkým nekonfigurujte server NFS na export súborového systému do neho samotného.
- Vyexportované systémy súborov by sa mali nachádzať vo svojich vlastných oddieloch. Zapisovaním do vyexportovaného súboru až do jeho zaplnenia môže útočník spôsobiť degradáciu systému, čo môže viesť k neprístupnosti systému súborov pre ostatné aplikácie alebo užívateľov, ktorí ho potrebujú.
- Nedovoľte klientom NFS vstupovať do systému súborov s povoľovacími údajmi koreňového alebo neznámeho užívateľa. Väčšinu implementácií NFS možno nakonfigurovať na mapovanie požiadaviek z privilegovaného alebo neznámeho do neprivilegovaného užívateľa, čím sa zabráni scenárom, podľa ktorých sa útočník snaží o prístup k súborom a vykonáva operácie súborov ako privilegovaný užívateľ.
- Nedovoľte klientom NFS spúšťať programy suid a sgid na exportovaných súborových systémoch, čím zabránite, aby klienti NFS spustili škodiaci kód s privilégiami. Ak útočník dokáže vytvoriť spustiteľný program, ktorý bude vlastniť privilegovaný vlastník alebo skupina, server NFS môže byť významným spôsobom poškodený. Uvedené možno vykonať zadaním voľby príkazu **mknfsmnt -y**.
- Použite zabezpečený NFS, ktorý používa na autentifikáciu hostiteľov zúčastňujúcich sa transakcií RPC šifrovanie DES. RPC je protokol, ktorý NFS používa na prenos požiadaviek medzi hostiteľmi. Zabezpečený NFS môže zmierniť pokusy útočníkov o oklamanie požiadaviek RPC zašifrovaním ich časovej značky. Prijímač úspešne dešifrujúci časovú značku a potvrdzujúci jej správnosť potvrdzuje, že požiadavka RPC prišla od dôveryhodného hostiteľa.
- Ak NFS nepotrebujete, vypnite ho, čím znížite počet možných útočných vektorov dostupných pre narušiteľa.

Okrem šifrovaní Triple DES a Single DES, NFS taktiež podporuje použitie šifrovania typu AES s autentifikáciou Kerberos 5. Opis konfigurácie protokolu Kerberos 5 na použitie typu šifrovania AES si prečítajte v príručke k správe systému NFS.

Súvisiace koncepty:

“Zabezpečenie NFS (Network File System)” na strane 265

Súvisiace informácie:

Kontrolný zoznam konfigurácie NFS
Spustiť démony NFS pri spustení systému
Konfigurácia servera NFS
Konfigurácia klienta NFS
Mapovanie identity
Export súborového systému NFS
nastavenie siete pre RPCSEC-GSS
Zrušenie exportu súborového systému NFS
Zmena exportovaného súborového systému
Prístup užívateľa root k exportovanému súborovému systému
Explicitné pripojenie súborového systému NFS
Automatické pripojenie podsystému
Vytvorenie preddefinovaných pripojení NFS
Odstránenie preddefinovaných pripojení NFS
súbor exportov pre NFS
príkaz mknfsmnt

Autentifikácia NFS (Network File System)

Systém NFS používa algoritmus DES na rôzne účely. NFS používa DES na šifrovanie časovej známky v správach volania vzdialenej procedúry (RPC) posielaných medzi servermi a klientmi NFS. Táto šifrovaná časová známka autentifikuje počítače práve tak, ako token autentifikuje odosielateľa.

Pretože NFS môže autentifikovať každú správu RPC vymieňanú medzi klientmi a servermi NFS, zabezpečí to dodatočnú, voliteľnú úroveň zabezpečenia pre každý súborový systém. Štandardne sú systémy súborov exportované so štandardnou autentifikáciou systému UNIX. Ak chcete využiť túto doplnkovú úroveň zabezpečenia, zadajte pri exporte súborového systému voľbu secure.

Kryptografia verejného kľúča pre zabezpečený systém NFS (Network File System):

Verejný kľúč i tajný kľúč užívateľa sú uložené a indexované podľa sieťového názvu v mape publickey.byname.

Súkromný kľúč je šifrovaný štandardom DES pomocou prihlasovacieho hesla užívateľa. Príkaz **keylogin** používa zašifrovaný súkromný kľúč, odšifruje ho pomocou prihlasovacieho hesla a odovzdá ho na zabezpečený lokálny server kľúčov, kde sa uloží pre použitie v ďalších transakciách RPC. Užívatelia nepoznajú svoje verejné a súkromné kľúče, pretože príkaz **yppasswd** okrem zmeny prihlasovacieho hesla generuje verejné a súkromné kľúče automaticky.

Démon keyserv je služba RPC, ktorá sa spúšťa na všetkých počítačoch NIS. **keyserv** spúšťa v rámci NIS tieto subrutiny verejných kľúčov:

- **key_setsecret**
- **key_encryptsession**
- podrutina **key_decryptsession**

Podprogram **key_setsecret** určuje serveru kľúčov, aby uložil tajný kľúč užívateľa (SK_A) pre ďalšie použitie. Štandardne je vyvolaný príkazom **keylogin**. Klientsky program vyvolá podprogram **key_encryptsession** na vytvorenie zašifrovaného kľúča pre komunikáciu, ktorý je zadaný v prvej transakcii RPC na server. Server kľúčov vyhľadá verejný kľúč na serveri a skombinuje ho s tajným kľúčom klienta (ktorý je nastavený predchádzajúcim podprogramom **key_setsecret**), aby vygeneroval spoločný kľúč. Server vytvorí požiadavku pre server kľúčov na dešifrovanie kľúča pre komunikáciu vyvolaním podprogramu **key_decryptsession**.

Vo volaniach týchto podprogramov je implicitne zahrnuté meno volajúceho, ktoré musí byť nejakým spôsobom autentifikované. Pre tento účel nemôže server kľúčov použiť autentifikáciu DES, pretože by sa vytvorilo blokovanie. Server kľúčov rieši tento problém ukladaním tajných kľúčov podľa ID užívateľa (UID) a povolením požiadaviek len

pre lokálne procesy root. Proces klienta potom spustí subrutinu **setuid** vo vlastníctve užívateľa s oprávneniami typu root, ktorá vykoná požiadavku na strane klienta a oznámi kľúčovému serveru reálny UID klienta.

Požiadavky autentifikácie NFS (Network File System):

Zabezpečená autentifikácia NFS je založená na schopnosti odosielateľa zašifrovať aktuálny čas, ktorý prijímač môže potom dešifrovať a porovnať so svojimi hodinami.

Tento proces má nasledovné požiadavky:

- Obaja agenti musia mať zhodný aktuálny čas.
- Odosielateľ a príjemca musia používať rovnaký šifrovací kľúč DES.

Dohodnutie aktuálneho času:

Ak sieť používa časovú synchronizáciu, démon timed bude udržiavať hodiny klienta a servera synchronizované. Ak nie, tak klient vypočíta správne časové známky na základe hodín servera.

Klient určí čas servera pred začatím relácie RPC a potom vypočíta časový rozdiel medzi vlastnými hodinami a hodinami servera. Klient potom upraví vlastnú časovú známku. V prípade, že počas relácie RPC dôjde k rozdielu medzi hodinami klienta a servera v takej miere, že server začne zamietať požiadavky klienta, klient opätovne určí čas servera.

Používanie rovnakého kľúča DES:

Klient a server vypočítajú ten istý šifrovací kľúč DES použitím kryptografie verejného kľúča.

Pre každého klienta A server B, kľúč s názvom *spoločný kľúč* môžu odvodiť len A a B. Tento kľúč je . Klient si odvodí spoločný kľúč výpočtom nasledovného vzorca:

$$K_{AB} = PK_B^{SK}A$$

kde *K* je spoločný kľúč, *PK* je verejný kľúč a *SK* je tajný kľúč a každý z týchto kľúčov je 128-bitové číslo. Server si odvodí zhodný spoločný kľúč výpočtom nasledovného vzorca:

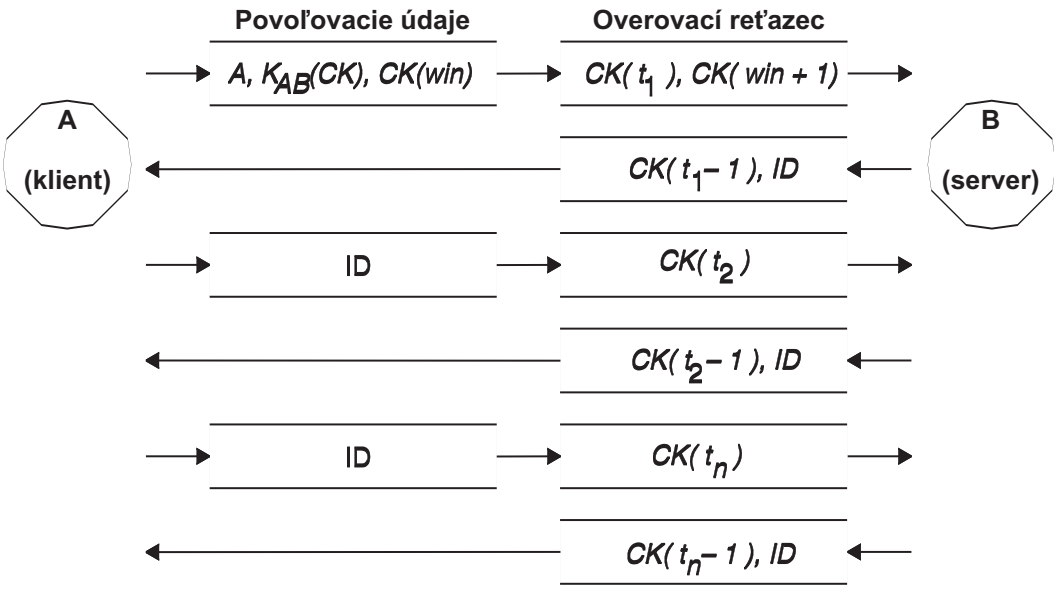
$$K_{AB} = PK_A^{SK}B$$

Spoločný kľúč môžu vypočítať len server a klient, pretože na jeho určenie je potrebný jeden alebo druhý tajný kľúč. Pretože spoločný kľúč má 128 bitov a DES používa 56-bitový kľúč, klient a server odčítajú 56 bitov zo spoločného kľúča na vytvorenie kľúča DES.

Proces autentifikácie NFS (Network File System):

Ak chce klient komunikovať so serverom, náhodne vygeneruje kľúč, ktorý sa používa na šifrovanie časových známok. Tento kľúč je známy ako *konverzačný kľúč* (*CK*).

Klient zašifruje kľúč pre komunikáciu pomocou spoločného kľúča DES (popísaného v časti Požiadavky na autentifikáciu) a odošle ho na server v prvej transakcii RPC. Tento proces je znázornený na nasledovnom obrázku.



Obrázok 15. Proces autentifikácie. Proces autentifikácie znázorňuje nasledovný obrázok.

Tento obrázok ukazuje klienta A pripájajúceho sa k serveru B. Výraz *K*(*CK*) znamená, že *CK* je šifrované so spoločným kľúčom DES *K*. Vo svojej prvej požiadavke oprávnenie RPC klienta obsahuje názov klienta (*A*), kľúč konverzácie (*CK*) a premennú s názvom *win* (window) šifrovanú s *CK*. (Predvolená veľkosť časového okna je 30 minút.) Overovanie klienta obsahuje v prvej požiadavke šifrovanú časovú známku a šifrované overenie zadaného časového okna, t. j. *win* + 1. Overenie časového okna sťažuje zistenie správneho oprávnenia a zvyšuje bezpečnosť.

Po autentifikácii klienta uloží server do tabuľky oprávnení nasledovné položky:

- Názov klienta, *A*
- Kľúč pre komunikáciu, *CK*
- Časové okno
- Časová známka

Server akceptuje len časové známky, ktoré sú chronologicky väčšie ako predchádzajúca zobrazená známka. Zabezpečí sa tým zamietnutie opakovaných transakcií. Server klientovi v overovači vráti ID indexu do tabuľky oprávnení, plus časovú známku klienta mínus 1, zašifrované pomocou *CK*. Klient vie, že takéto overenie mohol poslať len server, pretože len server pozná časovú známku odoslanú klientom. Dôvod, prečo sa z časovej známky odčíta 1, je zabezpečiť aby nebola platná a nemohla sa znova použiť ako overovač klienta. Po prvej transakcii RPC odošle klient serveru len svoje ID a zašifrovanú časovú známku a server odošle späť klientovu časovú známku mínus 1, zašifrovanú pomocou *CK*.

Pomenúvacie sieťové entity pre autentifikáciu DES

Autentifikácia DES používa pri vytváraní názvov sieťové názvy.

Sieťový názov je reťazec vytlačiteľných znakov určených na autentifikáciu. Verejné a súkromné kľúče sa neukladajú podľa mena užívateľa, ale podľa sieťového názvu. Mapa netid.byname NIS mapuje sieťový názov so zoznamom lokálnych UID a prístupových práv do skupín.

Mená užívateľov sú v rámci každej domény jednoznačné. Sieťové názvy sa určujú spájaním operačného systému a ID užívateľa s názvom domény služby NIS a internetovej domény. Bežným pravidlom pre vytváranie názvov domén je pripojiť názov internetovej domény (com, edu, gov, mil) k názvu lokálnej domény.

Sieťové názvy sa priraďujú počítačom aj užívateľom. Sieťový názov počítača sa vytvorí podobne ako v prípade užívateľa. Napríklad počítač s názvom hal v doméne eng.xyz.com má sieťový názov unix.hal@eng.xyz.com. Správna autentifikácia počítačov je dôležitá v prípade bezdiskových počítačov, ktoré vyžadujú plný prístup k domovským adresárom cez sieť.

Na autentifikáciu užívateľov z ľubovoľnej vzdialenej domény je potrebné vytvoriť pre nich položky v dvoch databázach NIS. Jedna s verejným a súkromným kľúčom, druhá s mapovaním lokálnych UID s prístupovými právami do skupín. Užívatelia vo vzdialenej doméne majú prístup ku všetkým lokálnym sieťovým službám, ako sú napríklad NFS a vzdialené prihlásenia.

Súbor /etc/publickey

Súbor /etc/publickey obsahuje mená užívateľov a verejné kľúče, na základe ktorých systém NIS vytvorí mapovanie publickey.

Mapa publickey sa používa na zabezpečenie siete. Každá položka v súbore pozostáva z mena užívateľa siete (ktoré označuje buď užívateľa alebo názov hostiteľa), za ktorým nasleduje verejný kľúč užívateľa (so zápisom v hexadecimálnej sústave), dvojbodka a užívateľom zašifrovaný tajný kľúč (rovnako so zápisom v hexadecimálnej sústave). Štandardne je jediným užívateľom v súbore /etc/publickey užívateľ nobody.

Na zmenu súboru /etc/publickey nepoužívajte textový editor, pretože súbor obsahuje šifrovacie kľúče. Ak chcete vykonať zmeny v súbore /etc/publickey, použite príkaz **chkey** alebo **newkey**.

Úvahy o zavedení systémov s verejnými kľúčmi

Keď po výpadku napájania reštartujete počítač, všetky uložené tajné kľúče budú odstránené a žiaden procese nebude mať prístup k zabezpečeným sieťovým službám, ako je napríklad NFS. Procesy typu root by mohli pokračovať v prípade, že existuje niekto, kto zadá heslo dešifrujúce tajný kľúč užívateľa s oprávneniami typu root . Riešením je ukladať tajný kľúč dešifrovaný užívateľom s oprávneniami typu root do súboru, ktorý môže server prečítať.

Nie všetky volania podprogramu **setuid** fungujú správne. Napríklad, ak podrutina **setuid** je volaná vlastníkom A a vlastník A sa neprihlásil do počítača odvtedy, čo bol spustený, tak podrutina nemôže mať prístup k žiadnym zabezpečeným sieťovým službám ako A. Väčšina volaní podrutiny **setuid** je vlastnená užívateľom s oprávneniami typu root a tajný kľúč užívateľa s oprávneniami typu root sa vždy ukladá v čase spustenia.

Úvahy o výkone zabezpečeného systému NFS (Network File System)

Zabezpečený NFS ovplyvní výkon systému niekoľkými spôsobmi.

- Klient aj server musia vypočítať spoločný kľúč. Výpočet spoločného kľúča trvá asi jednu sekundu. Takže vytvorenie prvého pripojenia RPC trvá asi 2 sekundy, pretože klient aj server musia túto operáciu vykonať. Po úvodnom pripojení RPC server kľúčov uloží do cache pamäte výsledky predchádzajúcich výpočtov, takže nemusí zakaždým vypočítať spoločný kľúč.
- Každá transakcia RPC vyžaduje nasledujúce operácie šifrovania DES:
  1. Klient zašifruje časovú známku požiadavky.
  2. Server ju odšifruje.
  3. Server zašifruje časovú známku odpovede.
  4. Klient ju odšifruje.

Keďže pri zabezpečenom NFS sa môže znížiť výkon systému, zvážte úžitok zvýšenej bezpečnosti s ohľadom na požiadavky výkonu systému.

Kontrolný zoznam zabezpečeného systému NFS (Network File System)

Tento kontrolný zoznam pomáha zaistiť správne fungovanie zabezpečeného NFS.

- Pri pripájaní systému súborov s voľbou **-secure** na klienta sa názov servera musí zhodovať s názvom hostiteľa servera v súbore /etc/hosts. Ak sa na rozlišovanie názvu hostiteľa používa názvový server, skontrolujte, či sa informácie o hostiteľovi, ktoré vrátil názvový server, zhodujú s položkou v súbore /etc/hosts. Ak sa tieto názvy nezhodujú, dôjde k chybám autentifikácie, pretože sieťové názvy pre počítače sú založené na primárnych položkách v súbore /etc/hosts a prístup ku kľúčom v mape **publickey** je na základe sieťového názvu.

- Nekombinujte zabezpečený a nezabezpečený export a pripojenie. V opačnom prípade môže byť prístup k súborom nesprávne určený. Napríklad, ak klientsky počítač pripojí zabezpečený systém súborov bez voľby **-secure**, alebo pripojí nezabezpečený systém s voľbou **-secure**, užívatelia budú mať prístup ako Nikto, a nie ako oni sami. Tento stav nastane aj vtedy, ak sa užívateľ, ktorý nie je známy pre systém NIS, pokúsi vytvoriť alebo upraviť súbory v zabezpečenom súborovom systéme.
- Pretože NIS musí po každom použití príkazov **chkey** a **newkey** rozšíriť novú mapu, tieto príkazy používajte len pri menšej záťaži siete.
- Neodstraňujte súbor /etc/keystore alebo súbor /etc/.rootkey. Ak reinštalujete, presúvate alebo vykonávate upgrade počítača, uložte súbory /etc/keystore a /etc/.rootkey.
- Dajte užívateľom pokyn, aby pre zmenu hesiel používali príkaz **yppasswd** namiesto príkazu **passwd**. Zabezpečuje to synchronizáciu hesiel a súkromných kľúčov.
- Keďže príkaz **login** nezíska znova kľúče z mapy publickey pre démona **keyserv**, užívateľ musí spustiť príkaz **keylogin**. Ak umiestnite príkaz **keylogin** do každého užívateľského súboru profile, príkaz sa spustí počas prihlasovania automaticky. Príkaz **keylogin** vyžaduje, aby užívatelia znova zadali svoje heslo.
- Keď generujete kľúče pre užívateľa s oprávneniami typu root na jednotlivých hostiteľoch pomocou príkazu **newkey -h** alebo **chkey**, musíte spustiť príkaz **keylogin** na postúpenie nových kľúčov procesu démon **keyserv**. Kľúče sa ukladajú v súbore /etc/.rootkey, ktorý je čítaný démonom **keyserv** vždy, keď je démon spustený.
- Pravidelne kontrolujte, či sú procesy typu démon **yppasswdd** a **ypupdated** na hlavnom serveri NIS spustené. Tieto procesy sú nevyhnutné na spravovanie mapy publickey.
- Pravidelne kontrolujte, či je na všetkých počítačoch používajúcich NFS spustený démon **keyserv**.

Konfigurácia zabezpečeného systému NFS (Network File System)

Ak chcete nakonfigurovať bezpečný systém NFS na hlavnom aj vedľajších serveroch NIS, vykonajte nasledujúce kroky.

1. Na hlavnom serveri NIS vytvorte pre každého užívateľa položku v súbore NIS /etc/publickey pomocou príkazu **newkey**:
   - Pre bežného užívateľa zadajte:

     ```
     smit newkey
     ```

     OR

     ```
     newkey -u username
     ```

     Pre užívateľa s oprávneniami typu root na hostiteľskom počítači zadajte:

     ```
     newkey -h hostname
     ```

   - Ďalšia možnosť je vytvoriť ich vlastné verejné kľúče pomocou príkazov **chkey** alebo **newkey**.
2. Vytvorte mapovanie NIS publickey. Príslušná mapa NIS publickey.byname je umiestnená len na serveroch NIS.
3. Zrušte komentár v nasledovných sekciách v súbore /etc/rc.nfs:

   ```
   #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. Spustite procesy typu démon **keyserv**, **ypupdated** a **yppasswdd** pomocou príkazu **startsrc**.

Na konfigurácia zabezpečeného systému NFS na klientoch NIS spustite proces typu démon **keyserv** pomocou príkazu **startsrc**.

Exportovanie súborového systému s použitím zabezpečeného systému NFS (Network File System)

Zabezpečený súborový systém NFS môžete exportovať vykonaním niektorého z nasledujúcich postupov.

- Pre export zabezpečeného systému súborov NFS s použitím SMIT vykonajte nasledujúce kroky:
  1. Skontrolujte, či NFS už beží, spustením príkazu **lssrc -g nfs**. Výstup indikuje, že démony nfsd a rpc.mountd sú aktívne.
  2. Skontrolujte, že existuje mapa publickey a že je spustený démon keyserv. Viac informácií nájdete v časti “Konfigurácia zabezpečeného systému NFS (Network File System)” na strane 271.
  3. Spustite rýchlu cestu **smit mknfsexp**.
  4. Zadajte príslušné hodnoty pre PATHNAME adresára, ktorý sa má exportovať, hodnotu režimu exportu adresára MODE a EXPORT directory now, system restart alebo obidve polia. V poli Use SECURE option zadajte yes.
  5. Zadajte ľubovoľné ďalšie voliteľné charakteristiky alebo akceptujte predvolené hodnoty.
  6. Ukončite SMIT. Ak súbor /etc/exports neexistuje, vytvorí sa.
  7. Kroky 3 až 6 zopakujte pre každý adresár, ktorý chcete exportovať.
- Pre export zabezpečeného systému súborov NFS s použitím textového editora vykonajte nasledujúce kroky:
  1. Súbor /etc/exports otvorte v textovom editore, ktorý bežne používate.
  2. Vytvorte položku každého adresára určeného na export tak, že použijete úplný názov cesty adresára. Zadajte každý adresár, ktorý chcete exportovať. Začnite na ľavom okraji. Adresár by nemal obsahovať iný adresár, ktorý už je exportovaný. Pozrite si dokumentáciu súboru /etc/exports kvôli opisu úplnej syntaxe pre položky v súbore /etc/exports, vrátane toho, ako sa má zadať voľba secure.
  3. Uložte a zatvorte súbor /etc/exports.
  4. Ak je systém NFS spustený, zadajte:
     ```
     /usr/sbin/exportfs -a
     ```
     Použitie voľby **-a** s príkazom **exportfs** odošle všetky informácie v súbore /etc/exports do jadra.
- Ak chcete exportovať súborový systém NFS dočasne (teda bez zmeny súboru /etc/exports), zadajte:
  ```
  exportfs -i -o secure /dirname
  ```
  kde dirname je názov súborového systému, ktorý chcete exportovať. Príkaz **exportfs -i** zadáva, že sa nemá hľadať zadaný adresár v súbore /etc/exports a že všetky voľby sa vezmú priamo z príkazového riadka.

Pripojenie systému súborov s použitím zabezpečeného systému NFS (Network File system)

Zabezpečený adresár NFS môžete pripojiť explicitne.

Pre explicitné pripojenie zabezpečeného adresára NFS vykonajte nasledujúce kroky:

1. Skontrolujte, či server NFS exportoval adresár pomocou príkazu:
   ```
   showmount -e ServerName
   ```
   kde ServerName je názov servera NFS. Tento príkaz zobrazí názvy adresárov, ktoré sú momentálne exportované zo servera NFS. Ak nie je uvedený adresár, ktorý chcete pripojiť, vyexportujte ho priamo zo servera.
2. Vytvorte lokálny bod pripojenia pomocou príkazu **mkdir**. Aby NFS úspešne vykonal pripojenie, musí byť k dispozícii adresár, ktorý bude fungovať ako bod pripojenia (alebo vlastník) pripojenia NFS. Tento adresár by mal byť prázdny. Tento bod pripojenia je možné vytvoriť podobne ako bežný adresár a nie sú potrebné žiadne špeciálne atribúty.
3. Skontrolujte, že existuje mapa publickey a že je spustený démon keyserv. Viac informácií nájdete v časti “Konfigurácia zabezpečeného systému NFS (Network File System)” na strane 271.
4. Napíšte
   ```
   mount -o secure ServerName:/remote/directory /local/directory
   ```

kde ServerName je názov servera NFS, /remote/directory je adresár na serveri NFS, ktorý chcete pripojiť a /local/directory je bod pripojenia na klientovi NFS.

Poznámka: Iba užívateľ s oprávneniami typu root môže pripojiť zabezpečený systém NFS.

Mapovanie podnikovej identity

Súčasné sieťové prostredia sú vytvorené zo zložitej skupiny systémov a aplikácií, čím vzniká potreba riadiť viaceré registre užívateľov. Práca s viacerými registrami užívateľov rýchlo narastá do rozsiahleho administratívneho problému, ktorý ovplyvňuje užívateľov, administrátorov a vývojárov aplikácií. Enterprise Identity Mapping (EIM) umožňuje administrátorom a vývojárom aplikácií predniesť tento problém.

Táto časť opisuje problémy, načrtáva súčasné priemyselné prístupy a vysvetľuje prístup EIM.

Riadenie viacerých užívateľských registrov

Mnohí administrátori riadia siete zahŕňajúce rôzne systémy a servery, z ktorých každý má jedinečný spôsob riadenia užívateľov prostredníctvom rôznych užívateľských registrov.

V týchto zložitých sieťach sú administrátori zodpovední za správu identít a hesiel každého užívateľa v rámci viacerých systémov. Okrem toho musia administrátori často tieto identity a heslá synchronizovať. Užívateľov zaťažuje nutnosť pamätať si viaceré identity a heslá a navzájom ich nezamieňať. Náklady na užívateľov a správcov v tomto prostredí sú veľké a správcovia často namiesto riadenia podniku trávia drahocenný čas odstraňovaním problémov pri neúspešných pokusoch o prihlásenie a resetovaním zabudnutých hesiel.

Problém správy viacerých registrov užívateľov takisto ovplyvňuje vývojárov aplikácií, ktorí chcú poskytovať viacvrstvové alebo heterogénne aplikácie. Dôležité obchodné údaje zákazníkov sú rozmiestnené v mnohých odlišných typoch systémov, pričom každý z nich vyžaduje vlastné registre užívateľov. V dôsledku toho musia vývojári v aplikáciách vytvárať špeciálne registre užívateľov a k nim priradené bezpečnostné sémantiky. Hoci sa tým problém pre vývojára aplikácie vyrieši, zaťaženie užívateľov a administrátorov sa napriek tomu zvyšuje.

Aktuálne prístupy k mapovaniu podnikovej identity

V súčasnosti je k dispozícii niekoľko prístupov priemyslu k riešeniu problému správy viacerých registrov užívateľov, no všetky poskytujú neúplné riešenia. Napríklad, protokol LDAP (Lightweight Directory Access Protocol) poskytuje distribuované riešenie registrov užívateľov. Ak chcú administrátori používať riešenia ako je LDAP, v konečnom dôsledku musia spravovať ďalší register užívateľov a bezpečnostnú sémantiku, prípadne za účelom používania týchto registrov nahradiť existujúce vytvorené aplikácie.

Pri používaní takýchto riešení musia administrátori spravovať viaceré bezpečnostné mechanizmy pre individuálne zdroje, čím sa zvyšuje administratívne zaťaženie a potenciálne narastá pravdepodobnosť vystavenia systémov bezpečnostným rizikám. Ak viaceré mechanizmy podporujú jeden zdroj, pravdepodobnosť zmeny oprávnenia pomocou jedného mechanizmu a nezmenenia oprávnenia pre jeden alebo viaceré z odlišných mechanizmov je oveľa vyššia. Napríklad, systém môže byť vystavený bezpečnostnému riziku vtedy, ak je užívateľovi odmietnutý prístup prostredníctvom jedného rozhrania, no daný prístup získa prostredníctvom jedného alebo viacerých odlišných rozhraní.

Po ukončení všetkých uvedených krokoch administrátori zaisťujú, že problém nie je úplne odstránený. Vo všeobecnosti platí, že podniky investujú príliš veľa finančných prostriedkov do aktuálnych registrov užívateľov a do ich priradenej bezpečnostnej sémantiky za účelom dosiahnutia praktického využitia daného riešenia. Vytvorenie ďalšieho registra užívateľa a priradenej bezpečnostnej sémantiky umožňuje vyriešiť problém na strane poskytovateľa aplikácie, no nie problémy užívateľov alebo administrátorov.

Ďalším riešením je použitie koncepcie jediného prihlásenia. Existuje niekoľko produktov, ktoré administrátorom umožňujú spravovať súbory obsahujúce všetky identity a heslá užívateľa. Tento prístup však má niekoľko slabých miest:

- Rieši len jeden problém, s ktorým sa užívatelia stretávajú. Hoci užívateľom umožňuje prihlásiť sa k viacerým systémom pomocou jednej identity a hesla, užívatelia musia napriek tomu disponovať heslami pre iné systémy, prípadne musia dané heslá spravovať.

- V súboroch sú uložené dešifrovateľné heslá alebo heslá zapísané obyčajným textom, čím vzniká bezpečnostné riziko a tým aj nový problém. Heslá by nemali byť nikdy uložené v súboroch s obyčajným textom, ani ľahko prístupné iným osobám, vrátane administrátorov.
- Nerieši problémy spojené s vývojármi nezávislých aplikácií, ktorí poskytujú heterogénne a viacvrstvové aplikácie. Títo vývojári musia pre svoje aplikácie poskytovať špeciálne registre užívateľov.

Napriek týmto nedostatkom niektoré podniky tieto riešenia používajú, pretože predstavujú určitý druh eliminovania problémov spojených s viacerými registrami užívateľov.

Použitie mapovania podnikovej identity

Architektúra EIM popisuje vzťahy medzi jednotlivcami alebo entitami (akými sú napríklad súborové servery a tlačové servery) v podnikovej sieti a mnohými identitami, ktoré ich zastupujú v rámci danej podnikovej siete. Táto architektúra okrem toho poskytuje množinu rozhraní API, ktoré aplikáciám umožňujú vytvárať dotazy týkajúce sa týchto vzťahov.

Napríklad, na základe užívateľskej identity určitej osoby v jednom registri užívateľa možno určiť, ktorá identita v inom registri užívateľa zastupuje rovnakú osobu. Ak užívateľ vykonal overenie pomocou jednej identity, ktorú možno mapovať k príslušnej identite v inom registri užívateľa, daný užívateľ nemusí znova poskytnúť oprávnenia za účelom overenia. Je potrebné vedieť len to, ktorá identita zastupuje daného užívateľa v príslušnom registri užívateľa. Znamená to, že architektúra EIM poskytuje v podnikovej sieti schematickú funkciu mapovania identít.

Schopnosť mapovať medzi identitami užívateľa v rôznych registroch prináša mnoho výhod. Aplikáciám poskytuje predovšetkým flexibilitu v používaní jedného registra pre autentifikáciu a úplne odlišného registra pre autorizáciu. Napríklad administrátor by mohol namapovať identitu SAP na prístup k prostriedkom SAP.

Mapovanie identity si vyžaduje, aby správcovia vykonali tieto kroky:

1. Vytvorenie identifikátorov EIM zastupujúcich osoby alebo entity v príslušnom podniku.
2. Vytvorenie definícií registrov EIM, ktoré popisujú existujúce registre užívateľov v danej podnikovej sieti.
3. Definovanie vzťahu medzi identitami užívateľov v registroch a vytvorenými identifikátormi EIM.

V existujúcich registroch nie je potrebné vykonávať žiadne zmeny kódu. Mapovania nie sú potrebné pre všetky identity v registri užívateľa. Architektúra EIM umožňuje mapovania typu ╚od jedného k viacerým┐ . (Inými slovami, jeden užívateľ s viac ako jednou identitou v jednom registri užívateľa). Architektúra EIM takisto umožňuje mapovania typu ╚od viacerých k jednému┐ . (Inými slovami, zdieľanie jednej identity v jednom registri užívateľa viacerými užívateľmi. Hoci je táto možnosť podporovaná, z bezpečnostných dôvodov sa neodporúča.) Administrátor môže v architektúre EIM zastupovať všetky typy registra užívateľa.

Architektúra EIM nevyžaduje kopírovanie existujúcich údajov do nového úložného priestoru a následnú synchronizáciu obidvoch kópií. Jediné nové údaje pridané architektúrou EIM sú informácie o vzťahoch. Administrátori tieto údaje spravujú v adresári LDAP, ktorý poskytuje flexibilitu v správe údajov na jednom mieste a vytvorení presných kópií na mieste, kde sa dané údaje používajú.

Kerberos

Kerberos je služba autentifikácie siete, ktorá poskytuje prostriedky na kontrolu identít principálov na fyzicky nie bezpečných sieťach. Kerberos poskytuje obojstrannú autentifikáciu, integritu údajov a súkromie za predpokladu, že prevádzka siete je náchylná na zachytenie, preskúmanie a nahradenie.

Principál Kerberos je jedinečná identita, ktorá používa služby autentifikácie Kerberos. Kerberos overuje identity a nespolieha sa na autentifikáciu hostiteľským operačným systémom, dôveru zakladá na hostiteľských adresách alebo vyžaduje fyzickú bezpečnosť všetkých hostiteľov v sieti.

Štítky Kerberos sú oprávnenia, ktoré overujú vašu totožnosť. Existujú dva typy štítkov: *štítok poskytujúci štítky* a *servisný štítok*. Štítok poskytujúci štítky je pre vašu úvodnú požiadavku totožnosti. Pri prihlasovaní do systému hostiteľa potrebujete niečo, čo overí vašu totožnosť, napríklad heslo alebo symbol. Keď máte štítok poskytujúci štítky, môžete potom použiť váš štítok poskytujúci štítky na požadovanie servisných štítkov pre konkrétne služby. Táto

metóda dvoch štítkov sa nazýva *dôveryhodná tretia strana* Kerberos. Váš štítok poskytujúci štítky vás autentifikuje serveru Kerberos a váš servisný štítok je vaším zabezpečeným úvodom do služby.

Dôveryhodná tretia strana alebo sprostredkovateľ sa v systéme Kerberos nazýva *Key Distribution Center* (KDC). KDC vydáva klientom všetky štítky Kerberos.

Prehľad zabezpečených vzdialených príkazov

Nasledujúce informácie poskytujú podrobnosti o zabezpečených vzdialených príkazoch.

Poznámky:

1. Od DCE (Distributed Computing Environment) verzie 2.2 môže bezpečnostný server DCE vracať štítky Kerberos verzie 5.
2. Všetky zabezpečené vzdialené príkazy (rcmds) používajú knižnicu Kerberos, verzia 5, poskytovanú službou IBM Network Authentication Service (NAS), ktorá je k dispozícii na disku Expansion Pack DVD. Musíte nainštalovať sadu súborov krb5.client.rte, ktorá je taktiež k dispozícii na disku Expansion Pack DVD.
3. Ak vykonávate migráciu operačného systému AIX pomocou médií DVD a knižnica Kerberos je už nainštalovaná, inštalačné skripty vás vyzvú k inštalácii sady súborov krb5.client.rte z disku Expansion Pack DVD.
4. Ak vykonávate migráciu operačného systému AIX pomocou prostriedkov NIM a knižnica Kerberos je už nainštalovaná, pridajte krb5 do adresára lpp_source.

Zabezpečené vzdialené príkazy (rcmds) sú: **rlogin**, **rcp**, **rsh**, **telnet** a **ftp**. Tieto príkazy sú súhrnne známe ako štandardná metóda autentifikácie AIX. Ďalšie metódy poskytuje knižnica Kerberos.

Pri použití metódy autentifikácie Kerberos Verzia 5 klient dostane štítok Kerberos Verzia 5 z bezpečnostného servera DCE alebo servera Kerberos. Štítok je časť aktuálneho DCE užívateľa alebo lokálnych oprávnení zašifrovaných pre server TCP/IP, ku ktorému sa chcú pripojiť. Démon na serveri TCP/IP zašifruje štítok. Táto akcia umožňuje serveru TCP/IP absolútne identifikovať užívateľa. Ak DCE alebo lokálny principál popísaný v štítku má povolený prístup k užívateľskému kontu operačného systému, vytvorí sa pripojenie. Bezpečnostné rcmds podporujú klientov a servery Kerberos z Kerberos Verzia 5 aj z DCE.

Okrem autentifikácie klienta, Kerberos Verzia 5 postupuje oprávnenia aktuálneho užívateľa na server TCP/IP. Ak sú splnomocnenia označené ako odosielateľné, klient ich odošle na server ako štítok poskytujúci štítky Kerberos. Ak na strane servera TCP/IP užívateľ komunikuje s bezpečnostným serverom DCE, démon rozšíri štítok poskytujúci štítky na úplné splnomocnenia DCE pomocou príkazu **k5dcecreds**.

Príkaz **ftp** používa inú metódu autentifikácie, ako ostatné bezpečnostné rcmds. Používa bezpečnostný mechanizmus GSSAPI na odovzdanie autentifikácie medzi príkazom **ftp** a démonom **ftpd**. Pomocou príkazov **clear**, **safe** a **private**, klient ftp podporuje šifrovanie údajov.

Medzi klientmi a servermi operačného systému, príkaz **ftp** umožňuje viacnásobné bajtové prenosy pre šifrované údajové pripojenia. Štandardy definujú len jednoduché bajtové prenosy pre šifrované údajové pripojenia. Pri pripojení k počítačom tretej strany a použití šifrovania údajov, príkaz **ftp** dodržiava obmedzenie jednoduchého bajtového prenosu.

Konfigurácia systému:

Pri všetkých bezpečnostných rcmds, mechanizmus konfigurácie na úrovni systému zisťuje, ktoré metódy autentifikácie sú pre tento systém povolené. Konfigurácia riadi odchádzajúce i prichádzajúce pripojenia.

Konfigurácia autentifikácie pozostáva z knižnice libauthm.a a príkazov **lsauthent** a **chauthent**, ktoré poskytujú prístup z príkazového riadka ku knižničným rutinám **get_auth_methods** a **set_auth_methods**.

Metóda autentifikácie definuje, ktorá metóda sa použije na autentifikáciu užívateľa cez sieť. Systém podporuje nasledovné metódy autentifikácie:

- Kerberos verzia 5 je najbežnejšia metóda, pretože je základom pre DCE.

- Kerberos Verzia 4 sa používa len pri bezpečnostných rcmds rlogin, rsh a rcp. Poskytuje sa na účely podpory kompatibility so staršími verziami iba na systémoch SP. Štítok Kerberos Verzia 4 sa neaktualizuje na oprávnenia DCE.

Ak je nakonfigurovaných viac metód autentifikácie a prvej metóde sa nepodarí pripojiť, klient sa pokúsi o autentifikáciu pomocou nasledujúcej nakonfigurovanej metódy autentifikácie.

Metódy autentifikácie možno konfigurovať v ľubovoľnom poradí. Jedinou výnimkou je, že štandardný AIX musí byť konečnou nakonfigurovanou metódou autentifikácie, pretože neexistuje ústupová voľba. Ak štandardný AIX nie je nakonfigurovanou metódou autentifikácie, nepokúsi sa o autentifikáciu hesla a všetky pokusy o pripojenie pomocou tejto metódy budú zamietnuté.

Môžete tiež nakonfigurovať systém bez akýchkoľvek metód autentifikácie. V takomto prípade systém zamietne všetky pripojenia z akéhokoľvek a do akéhokoľvek systému pomocou zabezpečených rcmds. Okrem toho, keďže Kerberos verzie 4 je podporovaný iba s príkazmi **rlogin**, **rsh**, a **rcp**, systém, ktorý je nakonfigurovaný iba na použitie Kerberos verzie 4, nepovoľuje pripojenia s použitím telnetu alebo FTP.

Overovanie platnosti užívateľa Kerberos verzie 5:

Metódu autentifikácie Kerberos verziu 5 môžete použiť na overenie platnosti užívateľa.

Pri použití metódy autentifikácie Kerberos Verzia 5, klient TCP/IP dostane servisný štítok šifrovaný pre server TCP/IP. Keď server dešifruje štítok, má bezpečnú metódu na identifikáciu užívateľa (pomocou DCE alebo lokálneho principála). Server však musí určiť, či je tomuto DCE alebo lokálnemu principála povolený prístup na lokálne konto. Mapovanie DCE alebo lokálneho principála na lokálne konto operačného systému sa spracúva pomocou zdieľanej knižnice libvaliduser.a, ktorá má jednu podrutinu, ktorá sa volá kvalid_user. Ak je preferovaná iná metóda mapovania, správca systému musí poskytnúť alternatívu pre knižnicu libvaliduser.a.

Konfigurácia DCE:

Pre používanie zabezpečených rcmds musia existovať dva principály DCE pre každé sieťové rozhranie, ku ktorému môžu byť pripojené.

Tie dva principály DCE sú:

```
host/FullInterfaceName
ftp/FullInterfaceName
```

kde *FullInterfaceName* je názov rozhrania a názov domény

Lokálna konfigurácia:

Pre používanie zabezpečených rcmds musia existovať dva lokálne principály pre každé sieťové rozhranie, ku ktorému môžu byť pripojené.

Tie dva lokálne principály sú:

```
host/CelýNázovRozhrania@NázovSféry
ftp/CelýNázovRozhrania@NázovSféry
```

kde *FullInterfaceName* je názov rozhrania a domény a *RealmName* je názov oblasti lokálneho Kerberos verzie 5.

Súvisiace informácie si pozrite v nasledujúcich zdrojoch:

- Podprogramy get_auth_method a set_auth_method v *Technical Reference: Communications, Volume 2*
- Príkaz chauthent v *Commands Reference, Volume 1*
- Príkaz lsauthent v *Commands Reference, Volume 3*

Autentifikácia v operačnom systéme AIX prostredníctvom služby Network Authentication Service alebo služieb iných ako AIX

Vo vydaniach starších ako AIX 6.1 zavádzací modul KRB5 spracovával autentifikáciu Kerberos v prostredí Network Authentication Service (NAS), kým zavádzací modul KRB5A spracovával autentifikáciu Kerberos v prostrediach iných ako AIX. Počnúc vydaním AIX 6.1, zavádzací modul KRB5 spracováva autentifikáciu Kerberos v prostredí Network Authentication Service (NAS) aj prostrediach iných ako AIX. Atribút **is_kadmind_compat** v súbore etc/security/methods.cfg určuje prostredie KRB5 alebo prostredie KRB5A. Počínajúc AIX 7.1 a ďalej nie je zavádzací modul KRB5A k dispozícii. Preto musíte atribút **is_kadmind_compat** použiť v súbore etc/security/methods.cfg buď na zadanie prostredia KRB5 alebo na zadanie prostredia KRB5A.

Keď je klient Kerberos nakonfigurovaný na autentifikáciu voči NAS, modul zavedenia KRB5 vykoná autentifikáciu Kerberos a riadenie principálov Kerberos. Modul umožňuje administrátorovi systému spravovať principálov Kerberos prostredníctvom príkazov na správu užívateľov systému AIX. Ak chcete použiť správu principálov, server Kerberos musí podporovať protokol administrácie kadmin. Služba NAS poskytuje túto podporu prostredníctvom démona **kadmind** (server Kerberos spustený v operačnom systéme AIX.

Poznámka: Pri konfigurácii klienta Kerberos musíte určiť, že autentifikácia sa vykonáva v službe NAS; v opačnom prípade sa klient nakonfiguruje pre autentifikáciu v službách iných ako AIX a správa principálov nebude k dispozícii.

Keď používate Kerberos v systéme inom ako AIX, principáli Kerberos sa ukladajú do systému iného ako AIX a nie je ich možné spravovať z operačného systému AIX pomocou rozhrania databázy Kerberos (kadmin). V tomto prípade musíte vykonať riadenie principálov oddelene pomocou nástrojov Kerberos na riadenie principálov. Tieto nástroje môžu byť súčasťou produktu Kerberos alebo môžu byť integrované do operačného systému (napríklad Windows 2000). Pôvodným účelom používania Kerberos v systémoch iných ako AIX bolo zabezpečiť autentifikáciu na serveroch Windows 2000 Active Directory, na ktorých sa správa principálov Kerberos vykonáva s použitím nástrojov a rozhraní API na správu kont Active Directory. Kerberos v systémoch iných ako AIX sa však môže používať v iných podporovaných KDC, v ktorých administračné rozhranie Kerberos nie je podporované.

Inštalácia a konfigurácia systému na integrované prihlásenie Kerberos pomocou IBM NAS:

Implementácia IBM Kerberos služieb sieťovej autentifikácie (NAS) sa dodáva v rozširujúcom balíku.

Ak chcete nainštalovať balík servera Kerberos verziu 5, nainštalujete sadu súborov krb5.server.rte spustením nasledujúceho príkazu:

```
installp –aqXYgd . krb5.server
```

Ak sa bude počítač, ktorý sa konfiguruje ako server Kerberos, používať aj ako klient Kerberos, nainštalujte celý balík Kerberos KRB5.

DCE má tiež sadu pomocných programov klienta Kerberos s rovnakými názvami ako pomocné programy Kerberos. Aby sa vyhlo kolíziám názvového priestoru medzi príkazmi DCE a Kerberos (t.j. medzi príkazmi **klist**, **kinit** a **kdestroy**), príkazy Kerberos sa inštalujú do adresárov /usr/krb5/bin a /usr/krb5/sbin.

Ak chcete spustiť príkazy Kerberos, musíte zadávať úplné názvy ciest príkazov, pokiaľ do vašej definície PATH nepridáte adresáre Kerberos nasledujúcim spôsobom:

```
export PATH=$PATH:/usr/krb5/sbin:/usr/krb5/bin
```

Poznámka: Java14 SDK tiež inštaluje príkaz **kinit** a ten môže byť v premennej prostredia PATH pred inými príkazmi **kinit**. Ak sú namiesto programu Java14 **kinit** potrebné príkazy služby sieťovej autentifikácie, program Java14 **kinit** presuňte na iné miesto vašej definície PATH.

Dokumentácia o službách Network Authentication Services sa dodáva v balíku krb5.doc.*lang*.pdf|html, kde *lang* predstavuje konkrétny podporovaný jazyk.

V operačnom systéme AIX sú k dispozícii dva databázové moduly na vytvorenie zloženého modulu zavedenia: LDAP a BUILTIN. Modul LDAP sa používa na prístup k informáciám uloženým v registri (adresári) LDAP a modul

BUILTIN na prístup k informáciám uloženým v registri súborov (lokálnom súborovom systéme). Vytvorený modul zloženého zavedenia sa obyčajne nazýva KRB5files alebo KRB5LDAP. Tieto názvy označujú, že KRB5 sa používa na autentifikáciu a lokálne súbory alebo na LDAP.

Služba sieťovej autentifikácie tiež podporuje ukladanie informácií Kerberos v lokálnom súborovom systéme (dedičnej databáze Kerberos) alebo v LDAP. Existujú štyri možné konfigurácie:

- KRB5files s informáciami o serveri Kerberos uloženými v dedičnej databáze Kerberos
- KRB5files s informáciami o serveri Kerberos uloženými v databáze Kerberos LDAP
- KRB5LDAP s informáciami o serveri Kerberos uloženými v dedičnej databáze Kerberos
- KRB5LDAP s informáciami o serveri Kerberos uloženými v databáze Kerberos LDAP

Keď sa ako mechanizmus na ukladanie princípálov Kerberos alebo informácií o užívateľoch a skupinách v systéme AIX používa LDAP, pred zavolaním konfiguračných príkazov Kerberos nakonfigurujte server LDAP. Keď nakonfigurujete LDAP, na konfiguráciu serverov Kerberos použite príkaz **mkkrb5srv**.

Konfigurácia servera služby sieťovej autentifikácie s úložným priestorom dedičnej databázy:

Pomocou príkazu **mkkrb5srv** môžete nastaviť KDC služby sieťovej autentifikácie a administračné servery s dedičnou databázou Kerberos a nakonfigurovať servery služby sieťovej autentifikácie.

Ďalšie informácie o používaní príkazu **mkkrb5srv** nájdete v príkaze **mkkrb5srv**.

Poznámka: Neinštalujte softvér servera Kerberos a DCE na ten istý fyzický systém. Ak to musíte urobiť, musíte zmeniť predvolené funkčné čísla internetových portov pre klientov a server DCE alebo pre klientov a server Kerberos. V oboch prípadoch môže táto zmena ovplyvniť interoperabilitu s existujúcimi rozmiestneniami DCE a Kerberos vo vašom prostredí. Informácie o koexistencii systémov DCE a Kerberos nájdete v dokumentácii o službách Network Authentication Services.

Kerberos Verzia 5 je nastavený tak, aby zamietal žiadosti o štítky od všetkých hostiteľov, ktorých hodiny nie sú v určenom maximálnom časovom odklone KDC. Predvolená hodnota pre maximálny časový odklon je 300 sekúnd (päť minút). Kerberos vyžaduje, aby medzi servermi a klientmi boli nakonfigurované nejaké formy časovej synchronizácie. Pre časovú synchronizáciu sa odporúča použiť démonov **xntpd** alebo **timed**. Ak chcete použiť démona **timed**, vykonajte nasledovné kroky:

1. Nastavte server KDC ako časový server spustením démona **timed**:
   ```
   timed -M
   ```
2. Spustite démona **timed** na každom klientovi Kerberos nasledujúcim spôsobom:
   ```
   timed -t
   ```
3. Pre konfiguráciu serverov Kerberos KDC a kadmin spustite príkaz **mkkrb5srv**. Ak napríklad chcete nakonfigurovať Kerberos pre oblasť MYREALM, server sundial a doménu xyz.com, spustite tento príkaz:
   ```
   mkkrb5srv -r MYREALM -s sundial.xyz.com -d xyz.com -a admin/admin
   ```

   Počkajte niekoľko minút, kým sa zo súboru /etc/inittab spustia príkazy **kadmind** a **krb5kdc**.

Služba sieťovej autentifikácie používa na ukladanie informácií priestor v súborovom systéme **/var**. Tieto informácie zahŕňajú súbory databázy, protokolu a pamäte cache splnomocnenia autentifikovaných užívateľov. Veľkosť týchto súborov sa môže časom zväčšovať. Pravidelným monitorovaním množstva voľného priestoru zabezpečte, aby mal súborový systém **/var** dostatok voľného priestoru na uchovávanie týchto informácií.

Nasleduje zvyčajný príkaz **mkkrb5srv**:

```
mkkrb5srv –r Realm_Name –s KDC_Server –d Domain_Name –a Admin_Name
```

Hodnoty premenných v Tabuľka 16 na strane 279 sú použité v nasledujúcom príklade, ako nakonfigurovať servery služby sieťovej autentifikácie pomocou dedičnej databázy.

Tabuľka 16. Názvy premenných príkazu **mkkrb5srv**

Názov premennej	Hodnota premennej
Názov oblasti	MYREALM
KDC Server	kdcsrv.austin.ibm.com
Názov domény	austin.ibm.com
Meno administrátora	admin/admin

Ak existuje konfigurácia servera Kerberos, môžete ju odstrániť pomocou príkazov **mkkrb5srv –U** alebo **unconfig.krb5**.

Upozornenie: Ak potrebujete zachovať existujúcu konfiguráciu servera Kerberos, nevykonajte nasledujúce kroky.

Nasledujúca procedúra je príkladom konfigurácie serverov služby sieťovej autentifikácie pomocou dedičnej databázy.

1. Zadajte tento príkaz:

   ```
   mkkrb5srv -r MYREALM -s kdcsrv.austin.ibm.com -d austin.ibm.com -a admin/admin
   ```

   Po zadaní tohto príkazu zadajte heslo hlavnej databázy.

   Vzhľadom na to, že služba sieťovej autentifikácie nepodporuje konfigurácie, v ktorých sa KDC a administračný server nachádzajú na rozdielnych hostiteľoch, pre KDC aj administračný server sa použije lokálny hostiteľ. Ak sa zobrazí táto chybová správa: Voľba -s nie je podporovaná, ignorujte ju.
2. Po výzve zadajte heslo hlavnej databázy.
3. Po výzve zadajte heslo administračného principála.

   Keď zadáte heslo administračného principála, príkaz **mkkrb5srv** spustí démonov **kadmind** a **krb5kdc** z cesty súboru /etc/inittab. Tento proces môže trvať niekoľko minút.
4. Spustením nasledujúcich príkazov overte položky v súbore /etc/inittab:

   ```
   lsitab krb5kdc
   lsitab kadm
   ```
5. Zadaním nasledujúceho príkazu overte, či boli servery KDC a kadmin spustené:

   ```
   ps -ef | grep -v grep | grep krb5
   ```

Príkaz **mkkrb5srv** vytvorí hlavný počítač KDC a administračné servery kadmin pre oblasť Kerberos (MYREALM). Vytvorí tiež konfiguračné súbory, inicializuje databázu principálov a spustí servery KDC a kadmin.

Vykonanie príkazu **mkkrb5srv** bude mať za následok nasledovné akcie:

1. Vytvorí sa súbor /etc/krb5/krb5.conf. Hodnoty pre názov sféry, server admin Kerberos a názov domény sa nastavia podľa špecifikácie v príkazovom riadku. Súbor /etc/krb5/krb5.conf tiež nastaví cesty pre protokolové súbory default_keytab_name, kdc a admin_server.
2. Vytvorí sa súbor /var/krb5/krb5kdc/kdc.conf. Súbor /var/krb5/krb5kdc/kdc.conf nastaví hodnoty pre premenné *kdc_ports*, *kadmin_port*, *max_life*, *max_renewable_life*, *master_key_type* a *supported_enctypes*. Tento súbor tiež nastaví cesty pre premenné *database_name*, *admin_keytab*, *acl_file*, *dict_file* a *key_stash_file*.
3. Vytvorí sa súbor /var/krb5/krb5kdc/kadm5.acl. Nastaví sa ovládanie prístupu pre principálov admin, root a host.
4. Vytvorí sa databáza a jeden principál admin. Budete vyzvaní nastaviť hlavný kľúč Kerberos a nazvať a nastaviť heslo pre identitu administratívneho principála Kerberos. Pre účely obnovy po havárii je rozhodujúce, aby boli hlavný kľúč a identita a heslo administračného principála bezpečne uložené.

Pre bližšie informácie si pozrite “Spúšťania vzorov” na strane 283 and “Chybové hlásenia a akcie obnovy” na strane 282.

Konfigurácia servera Kerberos s úložným priestorom LDAP:

Môžete nastaviť servery KDC a kadmin služby sieťovej autentifikácie na integrované prihlásenie Kerberos pomocou príkazu **mkkrb5srv**.

Hodnoty premenných v Tabuľka 17 sú použité v nasledujúcom príklade o konfigurácii komponentov servera služby sieťovej autentifikácie s úložným priestorom LDAP pomocou príkazu **mkkrb5srv**.

Tabuľka 17. Názvy premenných príkazu **mkrb5srv**

Názov premennej	Hodnota premennej
Realm_Name	MYREALM
KDC_Server	kdcsrv.austin.ibm.com
Domain_Name	austin.ibm.com
Admin_Name	admin/admin
server LDAP	kdcsrv.austin.ibm.com
Meno administrátora LDAP	cn=root
Heslo administrátora LDAP	tajné

Nasledujúca procedúra je príkladom konfigurácie komponentov servera služby sieťovej autentifikácie s úložným priestorom LDAP pomocou príkazu **mkkrb5srv**.

1. Spustite nasledujúci príkaz:
   ```
   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. Skontrolujte, či sa servery KDC a kadmind spustili zadaním nasledujúceho príkazu:
   ```
   ps -ef | grep -v grep | grep krb5
   ```

Spustenie príkazu **mkkrb5srv** s LDAP vytvorí podobné výsledky, ako spustenie príkazu s konfiguráciou dedičnej databázy. Pri použití LDAP sa však na lokálnom súborovom systéme nevytvoria databázy. Namiesto toho sa v súbore /var/krb5/krb5kdc vytvorí súbor .kdc_ldap_data na uchovanie informácií o LDAP.

Ďalšie informácie o používaní nájdete v príkaze **mkkrb5srv**.

Konfigurácia integrovaného prihlásenia Kerberos:

Keď dokončíte inštaláciu Kerberos, musíte nakonfigurovať systém na používanie protokolu Kerberos ako primárneho spôsobu autentifikácie užívateľov.

Ak chcete nakonfigurovať systémy, aby Kerberos ako primárny prostriedok autentifikácie užívateľa, spustite príkaz **mkkrb5clnt** s nasledovnými parametrami:

```
mkkrb5clnt -c KDC -r realm -a admin -s server -d domain -A -i database -K -T
```

Hodnoty premenných v časti Tabuľka 18 sú použite v nasledujúcom príklade znázorňujúcom konfiguráciu systému pre integrované prihlásenie Kerberos, pričom archívom užívateľov a skupín systému AIX je lokálny súborový systém.

Tabuľka 18. Názvy premenných príkazu **mkkrb5clnt**

Názov premennej	Hodnota premennej
Názov oblasti	MYREALM
KDC Server	kdcsrv.austin.ibm.com
Názov domény	austin.ibm.com
Administračný server	kdcsrv.austin.ibm.com
Meno administrátora	admin/admin
Databáza užívateľov a skupín systému AIX	files

Nasledujúci príkaz predstavuje príklad konfigurácie systému pre integrované prihlásenie Kerberos, pričom archívom užívateľov a skupín systému AIX je lokálny súborový systém.

Spustite nasledujúci príkaz:

```
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
```

Predošlý príklad prinesie nasledujúce akcie:

1. Príkaz vytvorí súbor /etc/krb5/krb5.conf. Hodnoty názvu oblasti, administračného servera Kerberos a názvu domény sú nastavené tak, ako je uvedené v príkazovom riadku. Cesty k protokolovým súborom default_keytab_name, kdc a kadmin sú tiež zaktualizované.
2. Príznak **-i** nakonfiguruje plne integrované prihlásenie. Zadanou databázou je umiestnenie, v ktorom sú uložené identifikačné informácie užívateľa AIX. Toto umiestnenie je iné ako úložný priestor principála systému Kerberos. Úložný priestor, kde sa ukladajú principály Kerberos, sa nastaví počas konfigurácie Kerberos.
3. Príznak **-K** nakonfiguruje Kerberos ako predvolenú schému autentifikácie. To umožní užívateľom, že v momente prihlásenia budú autentifikovaní so systémom Kerberos.
4. Príznak **-A** pridá položku do Databázy Kerberos za účelom vytvorenia užívateľa s oprávneniami typu root a užívateľa typu admin pre Kerberos.
5. Príznak **-T** získa štítok na poskytovanie štítkov administrátora servera.

Poznámka: Nepoužívajte voľbu **-D** príkazu **mkkrb5clnt** na konfiguráciu prostredia klienta Kerberos pre autentifikáciu v službe IBM Network Authentication Service (NAS). Ak neuvediete voľbu **-D** s príkazom **mkkrb5clnt**, atribút **is_kadmind_compat** nebude uvedený v súbore /usr/lib/security/methods.cfg a klientske prostredie Kerberos sa nakonfiguruje pre autentifikáciu v službe IBM NAS.

Overte konfiguráciu kontrolou súboru /etc/krb5/krb5.conf. Nasleduje príklad súboru /etc/krb5/krb5.conf na počítači klienta:

```
[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
```

Poznámka: Ak sa na úložný priestor principála Kerberos použije LDAP, súbor krb5.conf bude pod odsekom [realms] obsahovať nasledujúci riadok:

```
vdb_plugin_lib = /usr/lib/libkrb5ldplug.a
```

Ak je inštalovaný systém, ktorý sa nachádza v inej doméne DNS než KDC, musia byť vykonané nasledujúce dodatočné akcie:

1. Upraviť súbor /etc/krb5/krb5.conf a za [domain realm] pridať inú položku.
2. Namapovať inú doménu na vašu sféru.

Ak napríklad chcete zahrnúť klienta v doméne abc.xyz.com do vašej oblasti MYREALM, upravte súbor /etc/krb5/krb5.conf nasledujúcim spôsobom:

```
[domain realm]
     .austin.ibm.com = MYREALM
     .raleigh.ibm.com = MYREALM
```

Keď dokončíte konfiguráciu služby sieťovej autentifikácie, proces prihlásenia sa do operačného systému ostane nezmenený. Po úspešnom prihlásení budú mať užívatelia štítky na poskytovanie štítkov Kerberos priradené k spusteným procesom. Premenná prostredia $KRB5CCNAME užívateľa ukazuje na tento štítok na poskytovanie štítkov. Ak chcete overiť, či je prihlásenie úspešné a užívateľ má štítok poskytujúci štítky, použite príkaz **klist**.

Poznámka: Keď spustíte príkaz **mkkrb5clnt**, do súboru methods.cfg bude pridaný tento odsek.

```
KRB5:
        program = /usr/lib/security/KRB5
        program_64 = /usr/lib/security/KRB5_64
        options = is_kadmind_compat=yes

KRB5files:
        options = db=BUILTIN,auth=KRB5
```

Ďalšie informácie o:

- príkaze **mkkrb5clnt** nájdete v príkaze **mkkrb5clnt**.
- súbore methods.cfg nájdete v súbore methods.cfg.

Chybové hlásenia a akcie obnovy:

Medzi chyby, ktoré môžu nastať pri používaní príkazu **mkkrb5srv**, patria:

- Ak súbory krb5.conf, kdc.conf alebo kadm5.acl už existujú, príkaz **mkkrb5srv** neupraví hodnoty. Dostanete správu, že súbor už existuje. Všetky konfiguračné hodnoty možno meniť úpravou súborov krb5.conf, kdc.conf alebo or kadm5.acl.
- Ak niečo nesprávne napíšete a nevytvorí sa žiadna databáza, odstráňte vytvorené konfiguračné súbory a spustite príkaz znovu.
- V prípade nekonzistencie medzi databázou a konfiguračnými hodnotami odstráňte databázu z adresára /var/krb5/krb5kdc/* a príkaz spustite znova.
- Skontrolujte, či sú na vašom počítači spustení démoni **kadmind** a **krb5kdc**. Či sú títo démoni spustení, skontrolujete príkazom **ps**. Ak démoni neboli spustení, skontrolujte protokolový súbor.

Medzi chyby, ktoré môžu nastať pri používaní príkazu **mkkrb5clnt**, patria:

- Nesprávne hodnoty pre krb5.conf možno opraviť úpravou súboru /etc/krb5/krb5.conf.
- Nesprávne hodnoty pre príznak **-i** možno opraviť upravením súboru /usr/lib/security/methods.cfg.

Eliminácia závislosti na démonovi kadmind počas inej autentifikácie ako KRB5: Zavádzací modul KRB5 spôsobuje oneskorenie, keď démon kadmind nie je k dispozícii a keď používa mechanizmus inej autentifikácie ako KRB5, napríklad jednotné prihlásenie (SSO). Túto závislosť eliminuje nastavenie parametra kadmind_timeout v súbore **methods.cfg**.

Možné hodnoty sú kadmind_timeout=<seconds>, pričom hodnota seconds musí byť väčšia ako 0.

Keď sa zavádzací modul KRB5 pokúsi pripojiť k serveru kadmind, ktorý je vypnutý, dôjde k uplynutiu vyhradeného času pre TCP (Transmission Control Protocol). Parameter kadmind_timeout zamedzuje ďalšiemu oneskoreniu po úvodnom uplynutí vyhradeného času pre TCP. Parameter kadmind_timeout určuje časové okno pre pokus zavádzacieho modulu KRB5 o ďalšie pripojenie k serveru kadmind po úvodnom uplynutí vyhradeného času pre tcp. Keď je server kadmind spustený, stále je platné predvolené správanie.

Parameter kadmind_timeout je štandardne zakázaný. Ak chcete parameter kadmind_timeout povoliť, súbor methods.cfg zmeňte nasledovne:

```
KRB5:
        program = /usr/lib/security/KRB5
        options = kadmind_timeout=300
KRB5files:
        options = db=BUILTIN,auth=KRB5
```

Vytvorené súbory:

Príkaz **mkkrb5srv** vytvorí nasledovné súbory:

- /etc/krb5/krb5.conf
- /var/krb5/krb5kdc/kadm5.acl
- /var/krb5/krb5kdc/kdc.conf

Príkaz **mkkrb5clnt** vytvorí nasledovný súbor:

- /etc/krb5/krb5.conf

Voľba **mkkrb5clnt -i** *files* pridá do súboru /usr/lib/security/methods.cfg nasledovnú sekciu:

```
KRB5:
  program =
  options =
KRB5files:
  options =
```

Spúšťania vzorov:

Táto časť poskytuje príklady vzorových spúšťaní.

Nasleduje príklad príkazu **mkkrb5srv**:

```
# mkkrb5srv -r MYREALM -s sundial.xyz.com -d xyz.com -a admin/admin
```

Zobrazí sa výstup, podobný tomuto:

```
  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
```

Nasleduje príklad príkazu **mkkrb5clnt**:

```
mkkrb5clnt -r MYREALM -c sundial.xyz.com -s sundial.xyz.com \
           -a admin/admin -d xyz.com -i files -K -T -A
```

Zobrazí sa výstup, podobný tomuto:

```
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.
```

Eliminácia závislosti na démonovi kadmind počas autentifikácie:

Autentifikácia modulom zavedenia KRB5 môže zlyhať, keď nie je k dispozícii démon **kadmind**. Túto závislosť môžete eliminovať nastavením parametra *kadmind* v súbore methods.cfg.

Na vypnutie vyhľadávaní **kadmind** sú možné hodnoty kadmind=no alebo kadmind=false a na zapnutie vyhľadávaní **kadmind** sú možné hodnoty kadmind=yes alebo kadmind=true (predvolená hodnota je yes). Ak je táto voľba nastavená na hodnotu no, počas autentifikácie nebude kontaktovaný démon **kadmind**. Užívatelia sa preto môžu prihlásiť do systému bez ohľadu na stav démona **kadmind** za predpokladu, že pri vyzvaní zadajú správne heslo. Avšak pomocou príkazov na správu užívateľov v systéme AIX, ako napríklad **mkuser**, **chuser** a **rmuser**, nebude možné spravovať integrovaných užívateľov Kerberos, ak démon nie je k dispozícii (napríklad keď démon nie je spustený alebo počítač nie je dostupný).

Predvolená hodnota pre parameter *kadmind* je yes. To znamená, že počas autentifikácie sa vykonávajú vyhľadávania **kadmind**. V štandardnom prípade, ak démon nie je k dispozícii, autentifikácia môže trvať dlhšie.

Ak chcete zakázať kontrolu démona **kadmind** počas autentifikácie, modifikujte stanzy v súbore methods.cfg nasledovne:

```
KRB5:
        program = /usr/lib/security/KRB5
        options = kadmind=no

KRB5files:
        options = db=BUILTIN,auth=KRB5
```

Keď démon **kadmind** nie je dostupný, hlavný používateľ nebude môcť meniť používateľské heslá. V situáciách, ako sú napríklad prípady zabudnutého hesla, musíte démon **kadmind** sprístupniť. Taktiež, ak sa užívateľ rozhodne zadať meno principála Kerberos v prihlasovacom príkazovom riadku, primárny názov mena principála sa skráti (v súlade s obmedzením dĺžky mien užívateľov v systéme AIX). Toto skrátené meno sa použije na získanie informácií na identifikáciu užívateľa v systéme AIX (napríklad na získanie hodnoty domovského adresára.

Ak démon **kadmind** nie je dostupný (démon je nefunkčný alebo nie je prístupný), príkaz **mkuser** spôsobí nasledujúcu chybu:

```
Chyba 3004-694 pri pridávaní "krb5user": Nemáte potrebné oprávnenie.
```

Ak je parameter *kadmind* nastavený na hodnotu no alebo je démon **kadmind** neprístupný, systém nemôže overiť existenciu principála v databáze Kerberos, takže nezíska atribúty týkajúce sa Kerberos. Táto situácia spôsobuje neúplné alebo nepresné výsledky. Príkaz **lsuser** napríklad nemusí na dotaz ALL hlásiť žiadnych užívateľov.

Okrem toho bude príkaz **chuser** riadiť len atribúty týkajúce sa AIX a nie atribúty týkajúce sa Kerberos. Príkaz **rmuser** nevymaže principál Kerberos a príkaz **passwd** pre užívateľov autentifikovaných pomocou Kerberos zlyhá.

Ak je sieť, v ktorej sa démon **kadmind** nachádza, neprístupná, čas odozvy sa predĺži. Nastavením voľby kadmind na hodnotu no v súbore methods.cfg eliminujete oneskorenia počas autentifikácie, keď je počítač neprístupný.

Keď je démon **kadmind** vypnutý, užívatelia s heslami s ukončenou platnosťou sa nemôžu prihlásiť alebo zmeniť svoje heslá.

Ak nastavíte kadmind=no ale démon **kadmind** je spustený, môžete spustiť tieto príkazy: **login**, **su**, **passwd**, **mkuser**, **chuser** a **rmuser**.

Kerberos a služba NAS (Network Authentication Service): informácie o riešení problémov:

V tejto téme nájdete informácie o riešení problémov s klientmi Kerberos používajúcimi server Kerberos v operačnom systéme AIX.

Modul LDAP zapisuje informácie o chybách a ladení do podsystému syslog.

Služba sieťovej autentifikácie IBM používa na protokolovanie požiadaviek na démonov KDC a **kadmind** vlastné protokolové súbory. Protokolové súbory sú zadané v odseku [logging] súboru krb5.conf. Predvolené umiestnenia týchto súborov sú súbory /var/krb5/log/krb5kdc.log a /var/krb5/log/kadmin.log .

Ak sa problém týka IBM Tivoli Directory Server, skontrolujte protokolové súbory generované IBM Tivoli Directory Server. Štandardne sú protokolové súbory umiestnené v súboroch /var/ldap/ibmslapd.log a /var/ldap/db2cli.log.

- ***Ako môžem vytvoriť autentifikovaných užívateľov Kerberos v systéme AIX?***

  Užívateľ root musí získať splnomocnenia Kerberos, ktoré udeľujú požadované privilégium na vykonávanie administračných úloh. Administračné úlohy sa vykonávajú na tomto serveri KDC: kdcsrv.austin.ibm.com.

  Vytvorte užívateľské konto systému AIX (foo) a principála Kerberos (foo@MYREALM) v databáze Kerberos zadaním nasledujúcich príkazov:

  ```
  kinit root/kdcsrv.austin.ibm.com
  mkuser –R KRB5files SYSTEM=KRB5files registry=KRB5files foo
  ```

  Tieto príkazy tiež autentifikujú užívateľa pre súbory KRB5files.

  Ak ste nakonfigurovali protokol LDAP pomocou príkazu **mksecldap**, autentifikovaných užívateľov Kerberos môžete v systéme AIX vytvoriť zadaním nasledujúceho príkazu:

  ```
  mkuser –R KRB5LDAP SYSTEM=KRB5LDAP registry=KRB5LDAP foo
  ```

- ***Ako odstrániť užívateľa autentifikovaného pomocou Kerberos?***

  Ak chcete odstrániť užívateľa autentifikovaného pomocou Kerberos, zadajte tento príkaz:

  ```
  rmuser –R KRB5files foo
  ```

Ak ste nakonfigurovali LDAP pomocou príkazu **mksecldap**, zadaním nasledujúceho príkazu môžete odstrániť užívateľa autentifikovaného pomocou Kerberos:

```
rmuser –R KRB5LDAP foo
```

- ***Ako zmeniť heslo užívateľa autentifikovaného pomocou Kerberos?***

  Ak chcete zmeniť heslo užívateľa autentifikovaného pomocou Kerberos, zadajte tento príkaz:

  ```
  passwd –R KRB5files foo
  ```

- ***Čo sú rozšírené atribúty Kerberos v systéme AIX?***

  S informáciami o principáli Kerberos sa narába prostredníctvom rozšírených atribútov AIX pomocou príkazov AIX **lsuser** a **chuser**. Zobrazené môžu byť len atribúty, ktoré majú režim prístupu GET. K atribútom, ktoré majú režim prístupu SET, môže priradiť hodnotu privilegovaný užívateľ (užívateľ root v operačnom systéme AIX). Autentifikovaný užívateľ Kerberos v systéme AIX môže zobraziť svoje vlastné rozšírené atribúty Kerberos a iné povolené atribúty systému AIX, ako sú id, pgrp, groups, gecos, home a shell.

  V téme Tabuľka 19 nájdete zoznam rozšírených atribútov Kerberos v systéme AIX a ich režimy prístupu.

Tabuľka 19. Rozšírené atribúty Kerberos v systéme AIX a ich režimy prístupu

Názov rozšíreného atribútu	Opis	Režim prístupu
krb5_principal_name	Názov principála priradeného k menu užívateľa v systéme AIX.	GET
krb5_principal	Rovnaký ako atribút krb5_principal_name.	GET
krb5_realm	Názov oblasti Kerberos, do ktorej principál patrí.	GET
krb5_last_pwd_change	Čas poslednej zmeny hesla pre principál.	GET
krb5_attributes	Sada atribútov používaná KDC.	GET/SET
krb5_mod_name	Názov principála, ktorý naposledy upravil principál.	GET
krb5_mod_date	Čas poslednej zmeny principála.	GET
krb5_kvno	Verzia aktuálneho kľúča (hesla) principála.	GET/SET
krb5_mkvno	Číslo verzie hlavného kľúča databázy. Toto sa poskytuje kvôli kompatibilite s ostatnými implementáciami. Toto pole má hodnotu 0.	GET
krb5_max_renewable_life	Maximálna obnoviteľná životnosť akéhokoľvek štítka vydaného pre principál.	GET/SET
krb5_names	Zoznam párov name:hostname. Toto pole je určené pre použitie v budúcnosti. Neupravujte tento atribút.	GET/SET

Rozšírený atribút krb5_attributes reprezentuje sadu atribútov principála Kerberos, ktorú môže KDC používať. Privilegovaný užívateľ môže na úpravu týchto atribútov Kerberos používať príkaz **chuser**.

```
chuser –R KRB5files krb5_attributes=+requires_preauth krb5user
```

Ak chcete nastaviť príznak, pridajte pred neho znamienko plus (+). Ak chcete resetovať príznak, pridajte pred neho znamienko mínus (-). napríklad:

+*attribute_name* nastaví príznak

-*attribute_name* resetuje príznak

Poznámka: Keď sa vytvorí užívateľ, nastavia sa všetky atribúty s výnimkou týchto: requires_hwauth, needchange, password_changing_service a support_desmd5

Nasledujúci zoznam obsahuje atribúty pre rozšírený atribút krb5_attributes:

allow_postdated
: Ak je nastavený, pre principál sa môžu vydať postdatované štítky.

allow_forwardable
: Ak je nastavený, pre principál sa môžu vydať odosielateľné štítky.

allow_tgs_req
Ak je nastavený, pre principál budú vydané servisné štítky pomocou štítka poskytujúceho štítky.

allow_renewable
Ak je nastavený, pre principál sa môžu vydať obnoviteľné štítky.

allow_proxiable
Ak je nastavený, pre principál sa môžu vydať zastupiteľné štítky.

allow_dup_skey
Ak je nastavený, pre principál je povolená autentifikácia užívateľ - užívateľ.

allow_tix
Ak je nastavený, pre principál sa vydajú štítky.

requires_preauth
Ak je nastavený, pred vydaním štítka sa vyžaduje predbežná autentifikácia softvéru.

requires_hwauth
Ak je nastavený, pred vydaním štítka pre principál sa vyžaduje predbežná autentifikácia hardvéru pomocou softvéru.

needchange
Ak je nastavený, pred vydaním štítka sa musí zmeniť kľúč (heslo) pre principál

Poznámka: Ak je nastavený príznak needchange, užívateľ bude počas najbližšieho pokusu o prihlásenie vyzvaný na zmenu hesla. V tomto prípade je užívateľ autentifikovaný (pomocou Kerberos), ale mená štítok poskytujúci štítky. Na získanie štítka poskytujúceho štítky musí užívateľ vyvolať príkaz **kinit**. Príznak needchange sa používa len pre Kerberos, ktorý používa modul služieb sieťovej autentifikácie.

allow_svr
Ak je nastavený, pre principál sa môžu vydať servisné štítky.

password_changing_service
Ak je nastavený, principál je špeciálnym principálom pre službu zmeny hesla.

support_desmd5
Ak je nastavený, KDC môže vydávať štítky, ktoré používajú algoritmus kontrolného súčtu RSA MD5.

Poznámka: Nastavenie tohto atribútu môže spôsobiť problémy so vzájomnou spoluprácou.

- ***Ako môžem zobraziť zoznam rozšírených atribútov Kerberos v systéme AIX?***

  Ak chcete zobraziť zoznam rozšírených atribútov AIX Kerberos, spustite tento príkaz:

  ```
  lsuser –R KRB5files foo
  ```

  Pomocou voľby -a môžete tiež vypísať zoznam špecifických rozšírených atribútov. napríklad:

  ```
  lsuser -R KRB5files -f -a krb5_principal krb5_principal_name krb5_realm
  ```

- ***Ako môžem upraviť rozšírené atribúty Kerberos v systéme AIX?***

  Len privilegovaný užívateľ môže upraviť nasledujúce rozšírené atribúty s režimom prístupu SET: krb5_kvno, krb5_max_renewable_life, krb5_attributes a krb5_names.

  – Ak chcete zmeniť maximálnu obnoviteľnú životnosť pre akýkoľvek štítok vydaný pre užívateľa foo na päť dní, zadajte tento príkaz:

    ```
    chuser -R KRB5files krb5_max_renewable_life=432000 foo
    ```

  – Ak chcete zmeniť číslo verzie kľúča (hesla) principála priradeného k užívateľovi foo, zadajte tento príkaz:

    ```
    chuser -R KRB5files krb5_kvno=4 foo
    ```

  – Ak chcete nastaviť všetky atribúty principála Kerberos vypísané v Tabuľka 19 na strane 286, zadajte tieto príkazy:

    ```
    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
```

– Ak chcete resetovať všetky atribúty principála Kerberos vypísané v Tabuľka 19 na strane 286, zadajte tento príkaz:

```
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
```

– Ak chcete zmeniť hodnotu atribútu krb5_names a pridať dvojicu mena užívateľa a názvu hostiteľa systému AIX, zadajte nasledujúce príkazy:

```
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
```

- ***Ako zobraziť zoznam všetkých užívateľov definovaných v KRB5files?***

  Ak chcete zobraziť zoznam všetkých užívateľov autentifikovaných pomocou Kerberos, zadajte tento príkaz:

  ```
  lsuser -R KRB5files -a registry ALL
  ```

- ***Ako môžem skonvertovať užívateľa AIX na užívateľa autentifikovaného prostredníctvom Kerberos?***

  Pomocou príkazu **mkseckrb5** môžete skonvertovať užívateľa AIX na užívateľa autentifikovaného prostredníctvom Kerberos. Príkaz **mkseckrb5** skonvertuje neadministračných užívateľov (užívateľov s užívateľskými ID väčšími ako 201) na užívateľov autentifikovaných pomocou Kerberos. Ak vyvoláte príkaz **mkseckrb5**, budete musieť zadať názov a heslo administračného principála služby sieťovej autentifikácie. Ak nepoužívate voľbu randomizácie, musíte tiež zadať heslo pre každého konvertovaného užívateľa.

  Poznámka: Príkaz **mkseckrb5** konvertuje len lokálnych užívateľov. Užívateľov vo vzdialených doménach, napríklad LDAP, nie je možné pomocou tohto príkazu konvertovať.

  V nasledujúcom príklade sa *nepoužíva* voľba náhodného generovania počas konverzie užívateľa AIX na užívateľa autentifikovaného prostredníctvom Kerberos.

  1. Zadajte tento príkaz:

     ```
     mkseckrb5 foo
     ```

  2. Pred prihlásením užívateľa pomocou Kerberos nastavte atribúty SYSTEM a atribúty registra užívateľa nasledujúcim spôsobom:

     ```
     chuser -R KRB5files SYSTEM=KRB5files registry=KRB5files foo
     ```

  V nasledujúcom príklade sa používa voľba náhodného generovania počas konverzie užívateľa AIX na užívateľa autentifikovaného prostredníctvom Kerberos.

  1. Zadajte tento príkaz:

     ```
     mkseckrb5 -r user1
     ```

  2. Po dokončení konverzie nastavte atribúty SYSTEM a atribúty registra užívateľa a heslo nasledujúcim spôsobom:

     ```
     chuser -R KRB5files SYSTEM=KRB5files registry=KRB5files user1
     ```

     ```
     passwd -R KRB5files user1
     ```

- ***Ako zmeniť heslo pre principál Kerberos?***

  Koreňový užívateľ môže nastaviť heslo pre principál Kerberos zadaním nasledujúceho príkazu **passwd**:

  ```
  passwd -R KRB5files foo
  ```

  Ak zadáte príkaz **passwd**, zobrazí sa nasledujúca správa:

  ```
  Zmena hesla pre "foo"
  Staré heslo foo:
  Nové heslo foo:
  Zadajte nové heslo znovu:
  ```

  Ak zadáte príkaz **passwd** ako koreňový užívateľ, staré heslo bude ignorované. Výzvu na zadanie starého hesla môžete vypnúť pomocou voľby rootpwdrequired v súbore methods.cfg. Ak chcete vypnúť výzvu na zadanie starého hesla, upravte súbor /usr/lib/security/methods.cfg nasledujúcim spôsobom:

```
KRB5files:
   options = db=BUILTIN,auth=KRB5,rootrequiresopw=false
```

- ***Ako po úspešnom prihlásení získať štítok poskytujúci štítky, keď je nastavený atribút needchange?***

  Ak chcete po úspešnom prihlásení získať štítok poskytujúci štítky, keď je nastavený príznak needchange, vyvolajte príkaz **kinit**. Bližšie informácie o tejto téme nájdete v atribúte **needhange**.

- ***Prečo operačný systém AIX neakceptuje moje heslo?***

  Ak nie je vaše heslo akceptované, vykonajte nasledujúce kroky:

  – Skontrolujte, či sú servery KDC a kadmind spustené.

  – Uistite sa, že heslo spĺňa požiadavky operačného systému AIX a služby Network Authentication Service.

- ***Ako zmeniť pravidlá pre heslá?***

  Pravidlá pre heslá môžete v operačnom systéme AIX zmeniť úpravou atribútov politiky hesiel. Na zmenu politiky hesiel v databáze Kerberos môžete použiť nástroj kadmin servera sieťovej autentifikácie.

- ***Môže sa užívateľ autentifikovaný prostredníctvom Kerberos stať užívateľom autentifikovaným iba prostredníctvom štandardnej autentifikácie systémuAIX?***

  Užívateľ autentifikovaný prostredníctvom Kerberos (foo) sa môže autentifikovať prostredníctvom autentifikácie systému AIX funkciou **crypt()** nasledujúcim spôsobom:

  1. Nastavte heslo v systéme AIX pre užívateľa foo (/etc/security/passwd) pomocou príkazu **passwd**.
  2. Na účely testovania zvoľte iné heslo. napríklad:

     ```
     passwd -R files xy
     ```

  3. Zmeňte atribút SYSTEM užívateľa nasledujúcim spôsobom:

     ```
     chuser -R KRB5files SYSTEM=compat foo
     ```

     Zmenou atribútu SYSTEM zmeníte metódu autentifikácie z Kerberos na **crypt()**.

     Poznámka: Keďže užívateľ sa v tomto príklade prihlásil pomocou lokálnej autentifikácie, hodnota AUTHSTATE je kompatibilná a nevydá sa žiadny štítok poskytujúci štítky. Ak chcete používať autentifikáciu **crypt()** ako mechanizmus zálohovania, prejdite na krok 4.

  4. Ak chcete používať autentifikáciu **crypt()** ako mechanizmus zálohovania, zmeňte atribút SYSTEM nasledujúcim spôsobom:

     ```
     chuser -R KRB5files SYSTEM="KRB5files or compat" foo
     ```

- ***Ako zmeniť port kadmind klienta?***

  Démon **kadmind** sa používa na vykonávanie riadenia principálov Kerberos v systémoch autentifikovaných pomocou Kerberos, ktoré používajú NAS. Nasledujúci príklad znázorňuje zmenu portu **kadmind** klienta. V tomto príklade je démon **kadmind** spustený na serveri kdcsrv.austin.ibm.com a používa port 812.

  1. Príkaz **config.krb5** sa používa na konfiguráciu klienta:

     ```
     config.krb5 –C –r MYREALM –c kdcsrv.austin.ibm.com –s \
     kdcsrv.austin.ibm.com –d austin.ibm.com
     ```

  2. Upravte súbor krb5.conf a zmeňte číslo portu:

     ```
     admin_server = kdcsrv.austin.ibm.com:812
     ```

- ***Ako odstrániť splnomocnenia Kerberos?***

  Pri každom prihlásení užívateľa sa predchádzajúce splnomocnenia Kerberos prepíšu. Ak sa užívateľ odhlási, tieto splnomocnenia sa neodstránia. Ak chcete odstrániť tieto splnomocnenia, zadajte nasledujúci príkaz NAS **kdestroy**:

  ```
  /usr/krb5/bin/kdestroy
  ```

- ***Ako zmeniť životnosť štítka v KDC?***

  Ak chcete zmeniť životnosť štítka v KDC, vykonajte tieto kroky:

  1. Zmeňte atribút max_life v súbore kdc.conf . napríklad:

     ```
     max_life = 8h 0m 0s
     ```

  2. Zastavte a následne spustite démonov **krb5kdc** a **kadmind**.
  3. Zmeňte hodnotu max_life principálov krbtgt/MYREALM a kadmin/admin na hodnotu, ktorú ste zadali v kroku 1. Napríklad:

```
kadmin.local
kadmin.local: modify_principal -maxlife "8 hours" krbtgt/MYREALM
```

- ***Čo sa stane, ak démon kadmind nie je k dispozícii?***

  Ak démon kadmind nie je k dispozícii, autentifikácia môže trvať dlhšie alebo zlyhať. Autentifikácia môže zlyhať, ak je časť siete, v ktorej je démon kadmind umiestnený, neprístupná alebo ak je systém hosťujúci server kadmind vypnutý. Ak je systém neprístupný, nastavenie voľby kadmind v súbore methods.cfg na hodnotu no eliminuje oneskorenia počas autentifikácie.

  Keď je démon kadmind vypnutý, užívatelia sa nemôžu prihlásiť, ak uplynula platnosť ich hesiel. Ak démon kadmind nie je k dispozícii (démon je vypnutý alebo nedosiahnuteľný) a užívateľ zadá príkaz **mkuser**, zobrazí sa nasledujúca chyba:

  ```
  3004-694 Error adding "krb5user": Nemáte oprávnenie
  ```

  Okrem toho, príkazy **chuser** a **lsuser** riadia len atribúty týkajúce sa AIX, nie atribúty týkajúce sa Kerberos. Príkaz **rmuser** nevymaže principál Kerberos a príkaz **passwd** zlyhá pri užívateľoch autentifikovaných pomocou Kerberos.

  Ak démon kadmind nie je k dispozícii, koreňový užívateľ nemôže meniť heslá užívateľov. V prípade, ako napríklad zabudnuté heslo, musíte sprístupniť démona kadmind. Taktiež, ak sa užívateľ rozhodne zadať meno principála Kerberos v prihlasovacom príkazovom riadku, primárny názov mena principála sa skráti (v súlade s obmedzením dĺžky mien užívateľov v systéme AIX). Toto skrátené meno sa použije na získanie informácií na identifikáciu užívateľa v systéme AIX (napríklad na získanie hodnoty domovského adresára.

- ***Ako môžem nakonfigurovať operačný systém AIX na integrované prihlásenie Kerberos so správou užívateľov a skupín prostredníctvom protokolu LDAP v systéme AIX?***

  Ak plánujete používať LDAP na ukladanie informácií o užívateľoch a skupinách systému AIX, pred spustením príkazov **mkkrb5srv** a **mkkrb5clnt** pomocou príkazu **mksecldap** nakonfigurujte server a klienta LDAP. Na konfiguráciu serverov Kerberos sa používa príkaz **mkkrb5srv**. Na konfiguráciu klienta Kerberos sa používa príkaz **mkkrb5clnt** s voľbou LDAP -i. napríklad:

  ```
  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
  ```

- ***Ako sa používajú vzdialené príkazy povolené pomocou Kerberos po úspešnom prihlásení?***

  Keď sa užívateľ systému AIX prihlási do systému prostredníctvom Kerberos, pre vzdialené príkazy podporujúce Kerberos môže používať lístok udeľujúci lístky.

  V nasledujúcom príklade je server NAS nakonfigurovaný v kdcsrv.austin.ibm.com pomocou príkazu **mkkrb5srv**. Tento systém je tiež nakonfigurovaný pre prihlásenia založené na Kerberos pomocou príkazu **mkkrb5clnt**. Druhý systém, tx3d.austin.ibm.com, je nakonfigurovaný ako klient pomocou príkazu **mkkrb5clnt**.

  1. Uložte kľúče pre principál hostiteľa, host/tx3d.austin.ibm.com, do súboru /etc/krb5/krb5.keytab v systéme tx3d.
  2. Keďže ste na konfiguráciu počítača klienta použili **mkkrb5clnt**, tieto kľúče boli vyextrahované do súboru /var/krb5/security/keytab/tx3d.austin.ibm.com.keytab. Pripojte tento súbor k súboru /etc/krb5/krb5.keytab nasledujúcim spôsobom:

     ```
     ln -s /var/krb5/security/keytab/tx3d.austin.ibm.com.keytab /etc/krb5/krb5.keytab
     ```

  3. Ak je systém tx3d.austin.ibm.com nakonfigurovaný so serverom Kerberos iným ako AIX, explicitne vytvorte principál hostiteľa a extrahujte kľúče. napríklad:

     ```
     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:
     ```

     Vzhľadom na to, že je nástroj kadmin vyvolaný zo systému tx3d.austin.ibm.com, kľúče sa vyextrahujú do súboru /etc/krb5/krb5.keytab v systéme tx3d.austin.ibm.com. Tento krok môžete vykonať aj na počítači, ktorý hosťuje admin server Kerberos (napríklad kdcsrv). Keď vyextrahujete kľúče do súboru kľúčov, súbor sa presunie a zlúči so súborom /etc/krb5/krb5.keytab v tx3d.

  4. Ak chcete používať autentifikáciu Kerberos verziu 5 v systéme tx3d.austin.ibm.com, zapnite vzdialené príkazy:

```
lsauthent
Standard Aix
chauthent -k5 -std
lsauthent
Kerberos 5
Standard Aix
```

5. Ak chcete používať autentifikáciu Kerberos verziu 5 v systéme kdcsrv.austin.ibm.com, zapnite vzdialené príkazy:

```
chauthent -k5 -std
lsauthent
Kerberos 5
Standard Aix
```

6. Vytvorte užívateľa autentifikovaného pomocou Kerberos (foo) v kdcsrv a nastavte heslo.

```
mkuser -R KRB5files SYSTEM=KRB5files registry=KRB5files foo
passwd -R KRB5files foo
```

7. Vytvorte užívateľa foo v tx3d:

```
mkuser -R files foo
```

8. Použite protokol Telnet na systém kdcsrv.austin.ibm.com pomocou autentifikácie Kerberos.
9. Ak chcete zabezpečiť, aby bol vydaný štítok poskytujúci štítky, zadajte príkaz **klist**.

```
/usr/krb5/bin/klist
```

Nasledujú príklady vzdialených príkazov povolených pomocou Kerberos.

Poznámka: Pred spustením príkazov v nasledujúcich príkladoch odstráňte súbory .klogin, .rhost alebo hosts.equiv.

– Zadajte príkaz **date** vo vzdialenom hostiteľskom systéme tx3d.austin.ibm.com s príkazom **rsh**:

```
rsh tx3d date
```

– Prihláste sa do vzdialeného systému tx3d.austin.ibm.com pomocou príkazu **rlogin**:

```
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)
```

– Presuňte súbor do vzdialeného systému tx3d.austin.ibm.com pomocou príkazu **rcp**:

```
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"
Testovanie Kerberize-d rcp
```

– Použite protokol Telnet na vzdialený systém tx3d.austin.ibm.com so splnomocneniami Kerberos:

```
telnet tx3d
Pokúša sa...
Pripojený do tx3d.austin.ibm.com.
Znak zmeny významu je '^]'.
[ Kerberos V5 vás akceptuje ako "foo@MYREALM" ]
```

– Použite protokol Telnet na systém tx3d.austin.ibm.com a po vyzvaní zadajte názov hostiteľa a ID:

```
hostname
tx3d.austin.ibm.com
id
uid=234(foo) gid=1(staff)
```

– Pred použitím príkazu **ftp** povoleného pomocou Kerberos musíte použiť príkaz **kadmin** (z tx3d.austin.ibm.com) na vytvorenie principála služby FTP ftp/tx3d.austin.ibm.com a vyextrahovať ho do súboru /etc/krb5/krb5.keytab:

```
kadmin: addprinc -randkey ftp/tx3d.austin.ibm.com@MYREALM
kadmin: ktadd -k /etc/krb5/krb5.keytab ftp/tx3d.austin.ibm.com@MYREALM
kadmin:
```

Nasleduje príklad, ako sa FTP prihlasuje do vzdialeného systému tx3d.austin.ibm.com so splnomocneniami Kerberos.

```
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
```

Konfigurácia klienta Kerberos voči serveru Kerberos na systéme s výnimkou AIX:

Klienta Kerberos systému AIX môžete nakonfigurovať pre server Kerberos na nasledujúcich systémoch iných ako AIX: Windows Active Directory, Solaris a HP.

Konfigurácia protokolu Kerberos voči službe Kerberos servera Windows:

K dispozícii je niekoľko metód konfigurácie protokolu Kerberos voči službe Kerberos servera Windows.

Modul Kerberos len na autentifikáciu v KRB5 môžete použiť v časti autentifikácie modulu zloženého zavedenia. Počas konfigurácie zadáva užívateľ prostredie Kerberos pre modul zavedenia. Modul zavedenia KRB5 povoľuje Kerberos ako alternatívnu metódu autentifikácie voči službe Kerberos servera Windows 2000 alebo Windows 2003. Modul zavedenia AIX BUILTIN poskytuje prístup k funkciám bezpečnostnej knižnice. Modul zavedenia BUILTIN môžete skombinovať s modulmi zavedenia len na autentifikáciu s cieľom poskytnúť databázovú časť modulu zloženého zavedenia. Poskytuje tiež úložný priestor dedičných užívateľov a skupín a prístup k súborovému systému. Modul zavedenia LDAP môžete tiež použiť ako databázovú časť modulu zloženého zavedenia.

Na rozdiel od iných prostredí Kerberos so službou NAS v systéme AIX toto prostredie neposkytuje správu principálov Kerberos. Modul zavedenia KRB5 môžete použiť v prostrediach, v ktorých sú principáli Kerberos uložený v systéme inom ako AIX a nemôžu sa spravovať z operačného systému AIX prostredníctvom rozhrania **kadmin** databázy Kerberos. Riadenie principálov Kerberos sa vykonáva oddelene pomocou nástrojov riadenia principálov Kerberos. Tieto nástroje môžu byť súčasťou produktu Kerberos vyvinutého predajcami softvéru alebo môžu byť integrované do operačného systému, napríklad Windows 2000.

Konfigurácia služby Kerberos Windows Server 2000:

Služba Kerberos Windows Server 2000 a klient NAS sú schopné spolupráce na úrovni protokolu Kerberos (RFC1510). Keďže systém Windows Server 2000 nepodporuje rozhranie **kadmin**, počas konfigurácie klientov systému AIX musíte s príkazom **mkkrb5clnt** uviesť príznak –D. Na riadenie principálov v systémoch Windows sa používajú nástroje Windows.

Vykonaním nasledujúceho postupu môžete nakonfigurovať klienta systému AIX na autentifikáciu Kerberos v službe Kerberos systému Windows Server 2000.

1. Nastavte Windows Server 2000. Pozrite si dokumentáciu Microsoft týkajúcu sa konfigurácie servera Microsoft Active Directory.
2. Ak na klientovi systému AIX nie je nainštalovaný klient NAS, nainštalujte sadu súborov krb5.client.rte z disku AIX Expansion Pack.

3. Pri konfigurácii klienta Kerberos systému AIX použite príkaz **mkkrb5clnt** s nasledujúcimi konfiguračnými informáciami:

   realm Názov domény Windows Active Directory

   domain
   Názov domény počítača, ktorý je hostiteľom servera Active Directory

   KDC Názov hostiteľa servera Windows

   server Názov hostiteľa servera Windows

   Nasleduje príklad príkazu **mkkrb5clnt**:

   ```
   mkkrb5clnt -r MYREALM -d austin.ibm.com -c w2k.austin.ibm.com -s w2k.austin.ibm.com -D
   ```

   Voľba **-D** v príkaze **mkkrb5clnt** vytvorí voľbu **is_kadmind_compat=no** v súbore /etc/methods.cfg a nakonfiguruje prostredie klienta Kerberos na autentifikáciu voči iným systémom než AIX. Nepoužívajte voľbu **-D** príkazu **mkkrb5clnt** na konfiguráciu prostredia klienta Kerberos pre autentifikáciu v službe IBM Network Authentication Service (NAS).

   Poznámka: Keď spustíte príkaz **mkkrb5clnt**, do súboru methods.cfg bude pridaný tento odsek.

   ```
   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
   ```

   Bližšie informácie o:

   - príkaze **mkkrb5clnt** a povolených návestiach nájdete v príkaze **mkkrb5clnt**.
   - súbore methods.cfg nájdete v súbore methods.cfg.

4. Vzhľadom na to, že systém Windows podporuje typy šifrovania DES-CBC-MD5 a DES-CBC-CRC, zmeňte informácie o súbore krb5.conf tak, aby sa podobali nasledujúcim:

   ```
   [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. Vytvorte principál hostiteľa.

   Vzhľadom na to, že názvy kont Windows nemajú viacero častí, ako majú názvy principálov NAS, nemôžete vytvoriť konto priamo pomocou úplného názvu hostiteľa (host/<*fully_qualified_host_name*>). Namiesto toho sa inštancia principála vytvorí prostredníctvom mapovania názvu principála služby. V tomto prípade sa vytvorí konto, ktoré zodpovedá principála hostiteľa a pridá sa mapovanie názvu principála.

   Na serveri Active Directory vytvorte pomocou nástroja Active Directory Management nové užívateľské konto korešpondujúce s klientom tx3d.austin.ibm.com systému AIX nasledujúcim spôsobom:

   a. Vyberte zložku User.
   b. Kliknite pravým tlačidlom myši a vyberte možnosť New.
   c. Vyberte User.
   d. Zadajte tx3d v poli First name a kliknite na Next.
   e. Vytvorte heslo a kliknite na Next.
   f. Kliknutím na Finish vytvoríte principál hostiteľa.

6. Na počítači so systémom Windows Server 2000 zadajte príkaz **Ktpass** z príkazového riadka, čím vytvoríte súbor tx3d.keytab a nakonfigurujete hostiteľské konto systému AIX:

   ```
   Ktpass -princ host/tx3d.austin.ibm.com@MYREALM -mapuser tx3d -pass password -out tx3d.keytab
   ```

7. Skopírujte súbor tx3d.keytab na hostiteľský systém AIX.
8. Zlúčte súbor tx3d.keytab so súborom /etc/krb5/krb5.keytab v systéme AIX nasledujúcim spôsobom:

```
ktutil
rkt tx3d.keytab
wkt /etc/krb5/krb5.keytab
q
```

9. Vytvorte kontá domény Windows pomocou nástrojov riadenia užívateľov Active Directory.
10. Ak chcete vytvoriť kontá v systéme AIX korešpondujúce s kontami v doméne Windows a chcete používať autentifikáciu Kerberos, zadajte nasledujúci príkaz:

    ```
    mkuser registry=KRB5files SYSTEM=KRB5files foo
    ```

11. Ak sa chcete prihlásiť do systému AIX a overiť konfiguráciu, zadajte príkaz **telnet**.

    Nasleduje príklad integrovanej relácie prihlásenia Kerberos, ktorá používa KRB5 voči Windows Active Directory:

    ```
    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
    Pamäť cache štítka: FILE:/var/krb5/security/creds/krb5cc_foo@AUSTIN.IBM.COM_203
    Predvolený principál: foo@AUSTIN.IBM.COM

    Platný principál spúšťania služby Expires
    04/29/05 14:37:28 04/30/05 00:39:22 krbtgt/AUSTIN.IBM.COM@AUSTIN.IBM.COM
         Obnoviť do 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
    ```

Konfigurácia služby Windows Server 2003 Kerberos:

Klienta Kerberos môžete nakonfigurovať voči službe Windows Server 2003 Kerberos.

Ak chcete nakonfigurovať klienta so systémom AIX pre službu Kerberos na systéme Windows Server 2003, postupujte podľa pokynov v téme “Konfigurácia služby Kerberos Windows Server 2000” na strane 292.

Poznámka: Pomocný program **kpasswd** klienta NAS nemôže zmeniť heslo principála Kerberos v službe Windows Server 2003 Kerberos. Preto po prihlásení sa do systému AIX používajúceho protokol Kerberos užívateľ nemôže zmeniť heslo v systéme Windows Server 2003.

Konfigurácia protokolu Kerberos voči radičom domén Kerberos systémov Sun Solaris a HP-UX:

Klienta Kerberos môžete nakonfigurovať voči radičom domén Kerberos systémov Sun Solaris a HP-UX.

Na rozdiel od prostredia Kerberos so službou NAS v systéme AIX toto prostredie neposkytuje správu principálov Kerberos. Zavádzací modul KRB5 sa môže použiť v prostredí, v ktorom sú principáli Kerberos uložení v systéme inom ako AIX a nie je ich možné spravovať z operačného systému AIX pomocou rozhrania **kadmin** databázy Kerberos. Riadenie principálov Kerberos sa vykonáva oddelene pomocou nástrojov riadenia principálov Kerberos. Tieto nástroje môžu byť súčasťou produktu Kerberos vyvinutého predajcami softvéru alebo môžu byť integrované do operačného systému.

Konfigurácia Sun Solaris:

Klienta Kerberos môžete nakonfigurovať pre systém Sun Solaris.

Mechanizmus Sun Enterprise Authentication Mechanism (SEAM) a klient AIX NAS spolupracujú na úrovni protokolu Kerberos (RFC1510). Keďže rozhranie démona **kadmind** v systéme Solaris nie je kompatibilné s rozhraním **kadmin** klienta NAS systému AIX, pri konfigurácii klientov AIX uveďte s príkazom **mkkrb5clnt** príznak –D. Na riadenie principálov na systémoch Solaris sa používajú nástroje Solaris. Keďže sa na serveroch SEAM Kerberos a klientoch NAS systému AIX používa iný protokol zmeny hesla, zmena hesla principála spôsobí zlyhanie konfigurácie.

V nasledujúcom príklade sa používa Solaris.

Vykonaním nasledujúceho postupu môžete nakonfigurovať klienta systému AIX na autentifikáciu prostredníctvom Kerberos v mechanizme SEAM.

1. Nakonfigurujte SEAM pomocou dokumentácie Sun.
2. Ak na klientovi systému AIX nie je nainštalovaný klient NAS, nainštalujte sadu súborov krb5.client.rte z disku AIX Expansion Pack.
3. Pri konfigurácii klienta Kerberos systému AIX použite príkaz **mkkrb5clnt** s nasledujúcimi konfiguračnými informáciami:

   realm Názov oblasti Solaris Kerberos: AUSTIN.IBM.COM

   domain
   Názov domény počítača, ktorý je hostiteľom serverov Kerberos: Austin.ibm.com

   KDC Názov hostiteľa systému Solaris, ktorý je hostiteľom KDC: sunsys.austin.ibm.com

   server Názov hostiteľa systému Solaris, ktorý je hostiteľom démona **kadmin** (zvyčajne rovnaký ako KDC): sunsys.austin.ibm.com

   Poznámka: Keďže sú rozhrania **kadmin** klienta NAS systémov Solaris a AIX rozdielne, klienti NAS nepoužívajú názov servera a s príkazom **mkkrb5clnt** musíte uviesť príznak –D.

   Nasleduje príklad príkazu **mkkrb5clnt**:

   ```
   mkkrb5clnt -r AUSTIN.IBM.COM -d austin.ibm.com\
    -c sunsys.austin.ibm.com -s sunsys.austin.ibm.com -D
   ```

   Voľba **-D** príkazu **mkkrb5clnt** vytvorí voľbu **is_kadmind_compat=no** v súbore /etc/security/methods.cfg a nakonfiguruje prostredie klienta Kerberos pre autentifikáciu v iných systémoch ako AIX. Nepoužívajte voľbu **-D** príkazu **mkkrb5clnt** na konfiguráciu prostredia klienta Kerberos pre autentifikáciu v službe IBM Network Authentication Service (NAS).

   Poznámka: Keď spustíte príkaz **mkkrb5clnt**, do súboru methods.cfg bude pridaný tento odsek.

   ```
   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
   ```

   Bližšie informácie o:

   - príkaze **mkkrb5clnt** a povolených návestiach nájdete v príkaze **mkkrb5clnt**.
   - súbore methods.cfg nájdete v súbore methods.cfg.
4. Na vytvorenie principála hostiteľa host/tx3d.austin.ibm.com@MYREALM použite nástroj Solaris **kadmin** a uložte ho do súboru podobného nasledujúcemu:

   ```
   kadmin: add_principal -randkey host/tx3d.austin.ibm.com
   Principál "host/tx3d.austin.ibm.com@AUSTIN.IBM.COM" bol vytvorený.

   kadmin:ktadd -k /tmp/tx3d.keytab host/tx3d.austin.ibm.com
   ```

```
Položka pre princípál host/tx3d.austin.ibm.com s kvno 3,
    typ šifrovania DES-CBC-CRC bol pridaný do súboru kľúčov WRFILE:/tmp/tx3d.keytab.

kadmin: quit
```

5. Skopírujte súbor tx3d.keytab na hostiteľský systém AIX.
6. Zlúčte súbor tx3d.keytab so súborom /etc/krb5/krb5.keytab v systéme AIX nasledujúcim spôsobom:

```
ktutil
rkt tx3d.keytab
l
slot   KVNO   Principal
wkt /etc/krb5/krb5.keytab
q
```

7. Na vytvorenie princípála Kerberos použite nástroj **kadmin** systému Solaris.

```
add_principal sunuser
```

8. Ak chcete vytvoriť kontá v systéme AIX korešpondujúce s princípálom Kerberos v systéme Solaris a chcete používať autentifikáciu Kerberos, zadajte nasledujúci príkaz:

```
mkuser registry=KRB5files SYSTEM=KRB5files sunuser
```

9. Pomocou príkazu **telnet** sa prihláste do systému AIX s menom užívateľa sunuser a heslom a skontrolujte konfiguráciu.

   Nasleduje príklad integrovanej relácie prihlásenia Kerberos, ktorá používa KRB5 voči Solaris KDC:

```
telnet tx3d

echo $AUTHSTATE
KRB5files

echo $KRB5CCNAME
FILE:/var/krb5/security/creds/krb5cc_sunuser@AUSTIN.IBM.COM_207

Pozrite si splnomocnenia:
/usr/krb5/bin/klist
```

Konfigurácia HP-UX:

Klienta Kerberos môžete nakonfigurovať voči HP-UX.

Kroky na autentifikáciu v systéme HP-UX 11i sú podobné krokom v “Konfigurácia Sun Solaris” na strane 295. Rozhranie KDC systému HP-UX a klient NAS systému AIX spolupracujú na úrovni protokolu Kerberos (RFC1510). Protokol zmeny hesla je tiež kompatibilný. Keďže rozhranie démona **kadmind** v systéme HP-UX nie je kompatibilné s rozhraním **kadmin** klienta NAS systému AIX, pri konfigurácii klientov AIX musíte s príkazom **mkkrb5clnt** uviesť príznak –D.

Vykonaním nasledujúceho postupu môžete nakonfigurovať klienta systému AIX na autentifikáciu Kerberos v systéme HP-UX 11i s Kerberos, verzia 2.1.

1. Nakonfigurujte HP-UX 11i Kerberos verziu 2.1 pomocou dokumentácie HP.
2. Ak na klientovi systému AIX nie je nainštalovaný klient NAS, nainštalujte sadu súborov krb5.client.rte z disku AIX Expansion Pack.
3. Pri konfigurácii klienta Kerberos systému AIX použite príkaz **mkkrb5clnt** s nasledujúcimi konfiguračnými informáciami:

   realm Názov oblasti HP Kerberos: HPSYS.AUSTIN.IBM.COM

   domain
   Názov domény počítača, ktorý je hostiteľom serverov HP-UX Kerberos: austin.ibm.com

   KDC Názov hostiteľa systému HP-UX, ktorý je hostiteľom KDC: hpsys.austin.ibm.com

   server Názov hostiteľa servera HP-UX: hpsys.austin.ibm.com

Poznámka: Keďže sú rozhrania **kadmin** klienta NAS systémov HP-UX a AIX rozdielne, klienti NAS nepoužívajú názov servera a s príkazom **mkkrb5clnt** musíte uviesť príznak –D.

Nasleduje príklad príkazu **mkkrb5clnt**:

```
mkkrb5clnt -r AUSTIN.IBM.COM -d austin.ibm.com\
 -c hpsys.austin.ibm.com -s hpsys.austin.ibm.com -D
```

Voľba **-D** príkazu **mkkrb5clnt** vytvorí voľbu **is_kadmind_compat=no** v súbore /etc/security/methods.cfg a nakonfiguruje prostredie klienta Kerberos pre autentifikáciu v iných systémoch ako AIX. Nepoužívajte voľbu **-D** príkazu **mkkrb5clnt** na konfiguráciu prostredia klienta Kerberos pre autentifikáciu v službe IBM Network Authentication Service (NAS).

Poznámka: Keď spustíte príkaz **mkkrb5clnt**, do súboru methods.cfg bude pridaný tento odsek.

```
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
```

Bližšie informácie o:

- príkaze **mkkrb5clnt** a povolených návestiach nájdete v príkaze **mkkrb5clnt**.
- súbore methods.cfg nájdete v súbore methods.cfg.

4. Upravte súbor krb5.conf, aby sa typ šifrovania zhodoval s hodnotou použitou počas nastavenia HP-UX Kerberos (**krbsetup**). Ak sa použije hodnota DES-CRC, upravte odsek [libdefaults] v súbore krb5.conf na klientovi AIX nasledujúcim spôsobom:

   ```
   default_tkt_enctypes = des-cbc-crc

   default_tgs_enctypes = des-cbc-crc
   ```

5. Nástroj **kadmin_ui** systému HP-UX sa používa na vytvorenie principála hostiteľa host/tx3d.austin.ibm.com.
6. Vyextrahujte kľúč a uložte ho do súboru. Ak chcete vyextrahovať kľúče, z ponuky Edit v okne Principal Information vyberte voľbu Extract Service Key.
7. Skopírujte súbor tx3d.keytab na hostiteľský systém AIX.
8. Zlúčte súbor tx3d.keytab so súborom /etc/krb5/krb5.keytab v systéme AIX nasledujúcim spôsobom:

   ```
   ktutil
   rkt tx3d.keytab
   l
   slot   KVNO   Principal
   wkt /etc/krb5/krb5.keytab
   q
   ```

9. Na vytvorenie principála Kerberos užívateľa hpuser použite nástroj **kadmin_ui** systému HP-UX a následným kliknutím na záložku Edit/Attribute vymažete príznak pw_require.
10. Vytvorte konto systému AIX korešpondujúce s principálom Kerberos v systéme HP-UX nasledujúcim spôsobom:

    ```
    mkuser registry=KRB5files SYSTEM=KRB5files hpuser
    ```

11. Pomocou príkazu **telnet** sa prihláste do systému AIX s menom užívateľa hpuser a heslom a skontrolujte konfiguráciu.

    Nasleduje príklad integrovanej relácie prihlásenia Kerberos, ktorá používa KRB5 voči HP-UX:

    ```
    telnet tx3d

    echo $AUTHSTATE
    KRB5files

    Pozrite si splnomocnenia:
    /usr/krb5/bin/klist
    ```

12. Ak chcete zmeniť heslo, použite príkaz **passwd**.

Poznámka: Počas zmeny hesla bude vynútená politika hesiel HP-UX. Ak chcete určiť, ako nastaviť politiku hesiel, pozrite si dokumentáciu HP-UX.

Kerberos voči systémom s výnimkou AIX: otázky a informácie o odstraňovaní problémov:

Tu nájdete odpovede na otázky o klientoch Kerberos, ktorí používajú server Kerberos na systémoch s výnimkou AIX.

Poznámka: V nasledujúcich príkladoch sa používa Microsoft Active Directory Server. Tieto príklady však môžete použiť aj v systémoch Solaris a HP.

Ako prvý krok pri odstraňovaní problémov skontrolujte, či sú všetky servery a démony spustené.

Kerberos voči systémom s výnimkou AIX používa na zapisovanie informácií o chybách a ladení podsystém syslog. Bližšie informácie o protokolovaní syslog nájdete v démonovi **syslogd**.

- ***Ako vytvoriť užívateľa AIX?***

  Spustením nasledujúceho príkazu vytvoríte konto užívateľa AIX (foo):

  ```
  mkuser registry=KRB5files SYSTEM=KRB5files foo
  ```

  Príkaz **mkuser** vytvorí užívateľa v AIX. Okrem toho musíte pre užívateľa vytvoriť konto vo Windows Server Active Directory, ktoré zodpovedá kontu AIX. Vytvorením konta užívateľa vo Windows Server Active Directory sa implicitne vytvoria principály.

- ***Ako odstránim autentifikovaného užívateľa Kerberos?***

  Ak chcete odstrániť užívateľa autentifikovaného pomocou Kerberos, spustite tento príkaz:

  ```
  rmuser –R KRB5files foo
  ```

  Príkaz **rmuser** odstráni užívateľa z AIX. Pomocou nástrojov riadenia užívateľov Windows Server musíte tiež odstrániť užívateľa z Windows Active Directory.

- ***Ako zmeniť heslo užívateľa autentifikovaného pomocou Kerberos?***

  Ak chcete zmeniť heslo užívateľa autentifikovaného pomocou Kerberos, spustite tento príkaz:

  ```
  passwd –R KRB5files foo
  ```

  Ak KDC podporuje príkaz **kpasswd**, príkaz **passwd** zmení na serveri Kerberos heslo principála Kerberos foo@MYREALM.

- ***Ako povoliť užívateľom zmenu hesiel s ukončenou platnosťou na klientovi?***

  Ak chcete povoliť užívateľom zmenu hesiel s ukončenou platnosťou na klientovi, do súboru methods.cfg pridajte voľbu allow_expired_pwd=yes. Ak je táto voľba nastavená na hodnotu yes, užívatelia s heslami s ukončenou platnosťou budú vyzvaní na ich zmenu. Ak je voľba nastavená na hodnotu no alebo not present, užívateľov nie je možné autentifikovať.

  ```
  KRB5:
     program = /usr/lib/security/KRB5
     options = authonly,allow_expired_pwd=yes
  ```

- ***Ako skonvertovať užívateľa AIX na užívateľa autentifikovaného pomocou Kerberos?***

  Ak chcete skonvertovať užívateľa AIX na užívateľa autentifikovaného pomocou Kerberos, vykonajte nasledujúce kroky:

  1. Spustením nasledujúceho príkazu overte, či má užívateľ konto na Windows Server Active Directory:

     ```
     chuser registry=KRB5files SYSTEM=KRB5files foo
     ```

  2. Ak užívateľ nemá konto na Active Directory, vytvorte ho a nastavte atribút SYSTEM a atribút registra pomocou príkazu **chuser**. Konto Active Directory nemusí mať rovnaké meno užívateľa ako meno užívateľa AIX. Ak sa pre meno užívateľa AIX použije iné meno, na jeho mapovanie do mena Active Directory použite atribút auth_name.

     ```
     chuser registry=KRB5files SYSTEM=KRB5files auth_name=Christopher chris
     ```

- ***Čo robiť v prípade zabudnutého hesla?***

  Ak heslo zabudnete, administrátor Active Directory ho musí zmeniť. Koreňový užívateľ AIX nemôže nastaviť heslo principála Active Directory Kerberos.

- ***Aký je účel atribútov auth_name a auth_domain?***

  Poznámka: Tieto atribúty sú nepovinné. Ak systém AIX podporuje mená užívateľov dlhšie ako osem znakov, atribút auth_name možno nebudete potrebovať.

  Atribúty auth_name a auth_domain mapujú mená užívateľov AIX do názvov principálov Kerberos v KDC. Ak má napríklad užívateľ AIX Chris atribúty auth_name=christopher a auth_domain=SOMEREALM, názov principála Kerberos bude christopher@SOMEREALM. Pomocou atribútu auth_domain sa požiadavky namiesto do predvoleného názvu oblasti odošlú do názvu oblasti SOMEREALM. Užívateľovi, ktorý sa volá Chris, to umožní autentifikovať sa do oblasti SOMEREALM namiesto do oblasti MYREALM. V tomto príklade musí byť modifikovaný aj súbor krb5.conf, aby obsahoval názov oblasti SOMEREALM.

- ***Môže byť užívateľ autentifikovaný pomocou Kerberos autentifikovaný pomocou štandardnej autentifikácie AIX?***

  Áno, užívateľ autentifikovaný pomocou Kerberos môže byť autentifikovaný pomocou štandardnej autentifikácie AIX vykonaním nasledujúcich krokov:

  1. Nastavte heslo AIX (/etc/security/passwd) pomocou príkazu **passwd**:

     ```
     passwd -R files xy
     ```

  2. Zmeňte atribút registra a atribút SYSTEM nasledujúcim spôsobom:

     ```
     chuser -R KRB5files registry=files SYSTEM=compat foo
     ```

     Tento príkaz zmení autentifikáciu z Kerberos na compat (ktorá používa podprogram crypt). Pri najbližšom pokuse o prihlásenie užívateľa foo sa použije lokálne heslo zo súboru /etc/security/passwd.

     Autentifikáciu crypt môžete nasledujúcim spôsobom použiť aj ako mechanizmus zálohovania zmenou atribútu SYSTEM tak, aby povolil lokálnu autentifikáciu, keď autentifikácia pomocou Kerberos zlyhá:

     ```
     chuser -R KRB5files SYSTEM="KRB5files or compat" foo
     ```

- ***Je potrebné nastaviť server Kerberos v AIX pri používaní Windows Server 2000 Kerberos Service?***

  Nie, nemusíte nakonfigurovať server Kerberos (KDC) na klientovi AIX, pretože užívatelia sa autentifikujú voči Active Directory KDC. Ak plánujete používať AIX Network Authentication Service KDC ako server Kerberos na iný účel, musíte server Kerberos nakonfigurovať.

- ***Čo robiť, ak AIX neakceptuje heslo?***

  Ak AIX neakceptuje vaše heslo, vykonajte nasledujúce kroky:

  – Skontrolujte, či klient komunikuje s Windows 2000 Active Directory Server
  – Skontrolujte, či heslo spĺňa požiadavky AIX a Windows Server 2000 Active Directory. Pozrite si časť Change Show Policy, kde nájdete informácie o zmene pravidiel politiky hesiel v systéme AIX.

  Poznámka: Nemôžete zmeniť heslo pre Windows Server 2003 Kerberos Service.

- ***Čo robiť, ak sa nemôžete prihlásiť do systému?***

  Ak sa nemôžete do systému prihlásiť, urobte nasledovné:

  – V systéme Windows overte, či je KDC spustené vykonaním nasledujúcich krokov:
    1. Na ovládacom paneli vyberte ikonu administračných nástrojov.
    2. Vyberte ikonu služieb.
    3. Skontrolujte, či je centrum distribúcie kľúčov Kerberos v spustenom stave.
  – V systéme AIX overte, či súbor /etc/krb5/krb5.conf ukazuje na správne KDC a či má platné parametre.
  – V systéme AIX overte, či súbor kľúčov klienta obsahuje kľúč hostiteľa. Napríklad, ak je predvolený súbor kľúčov /etc/krb5/krb5.keytab, spustite nasledovné:

    ```
    ktutil
    rkt /etc/krb5/krb5.keytab
    l
    ```

  – Skontrolujte, či sa výstup príkazu **kvno**, ktorý je v súbore kľúčov, zhoduje s výstupom z príkazu **Ktpass**.
  – Skontrolujte, či atribúty auth_name a auth_domain, ak sú nastavené, odkazujú na platný názov principála v Active Directory KDC.
  – Skontrolujte, či je atribút SYSTEM nastavený na prihlásenie Kerberos.

– Skontrolujte, či neuplynula doba platnosti hesla.

- ***Ako vypnúť overovanie pomocou štítka poskytujúceho štítky?***

  Overovanie pomocou štítka poskytujúceho štítky môžete vypnúť zadaním voľby v súbore /usr/lib/security/methods.cfg pod odsekom KRB5 nasledujúcim spôsobom:

  ```
  KRB5:
     program = /usr/lib/security/KRB5
     options = tgt_verify=no
  KRB5files:
     options = db=BUILTIN,auth=KRB5
  ```

  Možné hodnoty pre voľbu tgt_verify sú no alebo false pre vypnutie overovania pomocou štítka poskytujúceho štítky a yes alebo true pre zapnutie overovania pomocou štítka poskytujúceho štítky. Štandardne je overovanie pomocou štítka poskytujúceho štítky zapnuté. Ak nastavíte voľbu tgt_verify na hodnotu no, overovanie pomocou štítka poskytujúceho štítky je vypnuté a nemusíte prenášať kľúče principála hostiteľa. Táto zmena len eliminuje potrebu súboru kľúčov na účely autentifikácie. Ostatné aplikácie povolené pomocou Kerberos môžu pre principály hostiteľa a služby vyžadovať súbor kľúčov.

- ***Čo robiť, ak sa nemôžem prihlásiť, lebo názov hostiteľa sa neinterpretuje a úplný názov hostiteľa zlyhá?***

  Overenie pomocou štítka poskytujúceho štítky vyžaduje vytvorenie principála host/<host_name> v KDC. Tento názov hostiteľa je úplným názvom klienta, na ktorom sa autentifikácia vykonáva. Systém klienta požaduje servisný štítok pomocou názvu principála hostiteľa host/<host_name>. V niektorých konfiguráciách nemôže počítač klienta získať úplný názov hostiteľa a namiesto neho dostane krátky názov. V takých prípadoch nastane nezhoda a zlyhá overovanie pomocou štítka poskytujúceho štítky a tiež prihlásenie. Ak má napríklad /etc/hosts len krátky názov a súbor /etc/netsvc.conf uvádza hosts=local,bind, rozlíšenie názvu vráti krátky názov.

  Ak chcete opraviť problémy rozlíšenia názvu, vykonajte niektorý z nasledujúcich krokov:

  – Upravte poradie rozlíšenia názvu v súbore /etc/netsvc.conf, aby sa vrátil úplný názov hostiteľa. Súbor netsvc.conf zadáva poradie postupnosti na rozlišovanie aliasov a názvov hostiteľa.

    V nasledujúcom príklade používa rozlišovač službu BIND na rozlíšenie názvu hostiteľa. Ak služba BIND zlyhá, rozlišovač použije súbor /etc/hosts. Ak obidve metódy zlyhajú, rozlišovače použijú nis.

    ```
    hosts=bind,local,nis
    ```

    Ak musí byť prvá metóda použitá v poradí vyhľadávania local, zmeňte krátky názov (myhost) v súbore /etc/hosts na úplný názov hostiteľa (myhost.austin.ibm.com).

  – Ak sa overenie pomocou štítka poskytujúceho štítky nevyžaduje, pokyny na jeho vypnutie nájdete v časti *Ako vypnúť overenie pomocou štítka poskytujúceho štítky?*.

- ***Prečo vracia podrutina passwdexpired hodnotu 0, keď platnosť hesla užívateľa Kerberos vypršala na serveri Kerberos inom ako AIX?***

  Podrutina **passwdexpired** vracia hodnotu 0, pretože informácie o uplynutí doby platnosti hesla nie je možné priamo získať zo servera Kerberos iného ako AIX kvôli nekompatibilite alebo nedostupnosti rozhraní **kadmin**.

  Príznak **allow_expired_pwd** v súbore **methods.cfg** umožňuje AIX získať informácie o uplynutí doby platnosti hesla s použitím autentifikačných rozhraní Kerberos. Skutočný stav informácií o uplynutí doby platnosti hesla sa získava počas prihlasovania alebo volaním podrutiny **authenticate** a podrutiny **passwdexpired**.

Modul Kerberos

Modul Kerberos je rozšírenínm jadra, ktoré používa klient NFS a serverový kód. Tento umožňuje klientovi NFS a serverovému kódu spracovať funkcie integrity správ Kerberos a súkromia bez volania démona **gss**.

Modul Kerberos je zavedený démonom **gss**. Použité metódy sú založené na verzii 1.2 služby sieťovej autentifikácie, ktorá je zase založená na MIT Kerberos.

Umiestnenie modulu Kerberos je: /usr/lib/drivers/krb5.ext.

Pre súvisiace informácie si pozrite démon **gss**.

Súvisiace informácie:

Zdroje informácií týkajúce sa služby IBM Network Authentication Service a súvisiacich technológií pre systém AIX v službe IBM developerWorks

Server RADIUS (Remote Authentication Dial-In User Service)

Služba Remote Authentication Dial-In User Service (RADIUS) spoločnosti IBM je protokol prístupu k sieti navrhnutý tak, aby vykonával autentifikáciu, autorizáciu a účtovanie. Ide o portový protokol, ktorý definuje komunikáciu medzi servermi prístupu k sieti (NAS), autentifikačnými servermi a servermi vykonávajúcimi správu používateľských kont.

NAS funguje ako klient RADIUS. Transakcie medzi klientom a serverom RADIUS sa autentifikujú prostredníctvom *zdieľaného kľúča*, ktorý sa cez sieť neposiela. Všetky používateľské heslá prenášané medzi klientom a serverom RADIUS sú šifrované.

Klient zodpovedá za odovzdávanie užívateľských informácií určeným serverom RADIUS a potom koná podľa vrátenej odpovede. Servery RADIUS zodpovedajú za príjem požiadaviek na pripojenie užívateľa, autentifikáciu užívateľa a za následné vrátenie všetkých klientom požadovaných konfiguračných informácií potrebných na dodanie služby užívateľovi. Keď sú nakonfigurované rozšírené proxy informácie, server RADIUS môže vo vzťahu k ostatným serverom RADIUS konať ako klient proxy. Server RADIUS používa ako protokol prenosu User Datagram Protocol (**UDP**).

Protokol autentifikácie a autorizácie RADIUS je založený na štandarde IETF RFC 2865. Server poskytuje aj protokol administrácie užívateľských kont definovaný v RFC 2866. Ostatnými podporovanými štandardmi sú RFC 2284 (EAP), časti RFC 2869, správy o uplynutí doby platnosti hesiel RFC 2882, MD5-Challenge a TLS. Bližšie informácie o týchto RFC nájdete v nasledujúcich odkazoch:

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

Všetky tieto štandardy RFC si môžete prezerať aj na http://www.ietf.org.

Inštalácia servera RADIUS

Server RADIUS môžete inštalovať buď pomocou príkazu **installp** alebo pomocou rozhrania SMIT. Softvér RADIUS sa nachádza na základnom inštalačnom médiu systému AIX, pričom názvy obrazov sú radius.base a bos.msg.<jazyk>.rte.

Ak LDAP directory plánujete používať ako svoju informačnú databázu na ukladanie užívateľských mien a hesiel, musíte nainštalovať ldap.server. Softvér **installp** musí byť nainštalovaný na každej inštalácii servera RADIUS.

Ak plánujete používať autentifikáciu EAP-TLS (napríklad na autentifikáciu digitálnych certifikátov v bezdrôtovej sieti), musíte nainštalovať aj OpenSSL 0.9.7 alebo novší a poskytnúť celú cestu do knižnice libssl.a v konfiguračnom súbore /etc/radius/radiusd.conf.

Démonov RADIUS môžete spustiť pomocou príkazu **radiusctl**. Po spustení bude na každom z nasledujúcich spustený jeden proces radiusd:

- autorizácia
- administrácia používateľských kont
- monitorovanie ostatných démonov

Pri opätovnom zavedení sa démony automaticky spustia na úrovni spustenia 2, pokiaľ nie je RADIUS nakonfigurovaný pre EAP-TLS.

Ak chcete zmeniť túto rutinu, upravte súbor /etc/rc.d/rc2.d/Sradiusd.

Poznámka: Ak je RADIUS nakonfigurovaný na autentifikovanie digitálnych certifikátov pomocou EAP-TLS, nie je možné nakonfigurovať démonov na automatické spustenie, pretože administrátor musí zadať certifikát passphrase, čo vyžaduje manuálne spustenie a reštartovanie RADIUS pomocou príkazu **radiusctl**.

Zastavenie a reštartovanie servera RADIUS

Démonov **radiusd** musíte zastaviť a reštartovať pri akýchkoľvek zmenách konfiguračného súboru /etc/radius/radiusd.conf servera RADIUS alebo predvolených autorizačných súborov /etc/radius/authorization/default.policy alebo /etc/radius/authorization/default.auth, čo môžete vykonať zo SMIT alebo z príkazového riadka.

Na spustenie, reštartovanie a zastavenie servera RADIUS sa používajú tieto príkazy:

```
radiusctl start
radiusctl restart
radiusctl stop
```

Zastavenie a spustenie servera RADIUS je nevyhnutné, pretože démon musí vytvoriť pamäťovú tabuľku všetkých predvolených atribútov, ktoré obsahujú vyššie uvedené konfiguračné súbory. Zdieľaná pamäť sa používa pri každom lokálnom užívateľovi a tabuľka lokálnych užívateľov sa z výkonnostných dôvodov vytvára až počas inicializácie démonov.

Funkcia On-demand:

Podľa potreby môžete spustiť viacerých démonov autentifikačného a evidenčného servera RADIUS.

Každý server načúva na samostatnom porte. Súbor radiusd.conf sa dodáva s predvoleným číslom portu 1812 pre autentifikáciu a 1813 pre administráciu používateľských kont. Tieto čísla portov sú vyhradené pre znakovú sadu IANA. Po aktualizácii súboru radiusd.conf možno tieto porty podľa potreby používať spolu s inými portmi (viacerými). Dbajte, aby ste nepoužili čísla portov, ktoré sú už priradené iným službám. Keď v súbore radiusd.conf zadáte do polí **Authentication_Ports** a **Accounting_Ports** viacero čísel portov, démon **radiusd** sa spustí pre každý port. Démoni budú načúvať na svojom príslušnom čísle portu.

Konfiguračné súbory RADIUS

Démon RADIUS používa niekoľko konfiguračných súborov. Vzorové verzie týchto súborov sa dodávajú v pakete RADIUS.

Všetky konfiguračné súbory vlastní užívateľ root a skupina security. Pomocou SMIT (System Management Interface Tool) môžete upravovať všetky konfiguračné súbory s výnimkou súboru slovníka. Predtým, ako sa prejavia akékoľvek úpravy v konfiguračných súboroch, sa server musí reštartovať.

Súbor radiusd.conf:

Súbor radiusd.conf obsahuje konfiguračné parametre pre RADIUS.

RADIUS štandardne hľadá súbor radiusd.conf v adresári /etc/radius. Položky konfiguračného súboru musia mať formáty, ktoré sú ukázané v súbore. RADIUS akceptuje len platné kľúčové slová a hodnoty a predvolenú hodnotu použije vtedy, keď sa nepoužije platné kľúčové slovo alebo hodnota. Po spustení démonov RADIUS skontrolujte výstup SYSLOG kvôli chybám konfiguračných parametrov. Všetky chyby konfigurácie nevedú k zastaveniu servera.

Tento súbor by mal byť náležite chránený proti prečítaniu a proti zápisu, pretože má vplyv na chovanie autentifikačných a evidenčných serverov. Okrem toho môže tento súbor obsahovať aj dôverné údaje.

Dôležité: Ak súbor radiusd.conf upravujete, nemeňte poradie jeho položiek. Panely SMIT závisia od poradia.

Nasleduje príklad súboru radiusd.conf:

```
#-------------------------------------------------------------------#
#              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     :
```

Metódy autentifikácie EAP pre každého užívateľa možno nastaviť pomocou SMIT. Ak chcete nastaviť metódy EAP pre každého užívateľa, vykonajte tieto kroky:

```
Radius Server
     -> Configure users
          -> Local Database
             Adresár LDAP
                  -> Add a user
                     Change/Show Characteristics of a user
                          ->
                          Login User ID [ ]
                          EAP Type [0 2 4]
                          Password Max Age
```

Ak vyberiete typ EAP, k dispozícii budú tieto výbery:

0 Žiadna

2 MD5 - Challenge

4 TLS

Vybratá metóda EAP sa porovnáva so sekvenciou metódy autentifikácie nastavenou v súbore radiusd.conf na vykonanie autentifikácie.

Súbor /etc/radius/clients:

Súbor clients obsahuje zoznam klientov, ktorí majú povolené vykonávať požiadavky servera RADIUS.

Pre každého klienta NAS alebo AP musíte zvyčajne zadať IP adresu klienta spolu so zdieľaným kľúčom medzi serverom RADIUS a klientom a voliteľný *poolname* pre IP Pooling.

Súbor obsahuje položky v nasledujúcej forme:

```
<Client IP Address>    <Shared Secret>    <Pool Name>
```

Vzorový zoznam položiek bude vyzerať nasledovne:

```
10.10.10.1      mysecret1      floor6
10.10.10.2      mysecret2      floor5
```

Zdieľaný kľúč je znakovým reťazcom, ktorý je nakonfigurovaný na klientskom hardvéri aj na serveri RADIUS. Zdieľaný kľúč môže mať maximálnu dĺžku 256 bajtov a rozlišuje veľké a malé písmená. Zdieľaný kľúč sa nikdy neposiela v žiadnom z paketov RADIUS a nikdy sa neposiela cez sieť. Správcovia systému sa musia presvedčiť, či je správny kľúč nakonfigurovaný na obidvoch stranách (na strane klienta a na strane servera RADIUS). Zdieľaný kľúč sa používa na šifrovanie informácií o užívateľskom hesle a pomocou atribútu Message Authentication sa môže použiť na overenie celistvosti správ.

Každý zdieľaný kľúč klienta by mal byť v súbore /etc/radius/clients jedinečný a ako pri každom dobrom hesle je najlepšie, keď sa v ňom použije zmes veľkých/malých písmen, číslic a symbolov. Ak chcete zachovať bezpečnosť zdieľaného kľúča, vytvorte ho aspoň so 16 znakmi. Súbor /etc/radius/clients sa dá upraviť pomocou SMIT. Zdieľaný kľúč by sa mal často meniť, aby sa zamedzilo útokom na slovníky.

poolname je názov oblasti, z ktorej sa počas dynamického prekladu alokujú globálne IP adresy. Správca systému vytvorí *poolname* pri nastavovaní servera RADIUS. S použitím panela SMIT bude *poolname* pridaná z **Configure Proxy Rules > IP Pool > Create an IP Pool**. Používa sa počas zhromažďovania IP na strane servera.

Súbor /etc/radius/dictionary:

Súbor dictionary obsahuje popisy atribútov, ktoré sú definované protokolom RADIUS a podporované serverom AIX RADIUS.

Používa ho démon RADIUS pri overovaní a vytváraní údajov paketu. Aj špecifické atribúty predajcu by sa mali vkladať sem. Slovníkový súbor možno upravovať pomocou akéhokoľvek editora. Nie je tu žiadne rozhranie SMIT.

Nasleduje časť vzorového slovníkového súboru:

```
#########################################################################
#                                                                       #
#   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
.
.
.
```

Poznámka: Akýkoľvek atribút, ktorý je definovaný v súbore default.policy alebo súbore default.auth (alebo pre konkrétny súbor user_id.policy alebo user_id.auth) musí byť platným atribútom RADIUS definovaným v konfiguračnom súbore slovníka systému AIX. Ak sa niektorý atribút v tomto slovníku nenachádza, démon **radiusd** sa nezavedie a zaznamená sa chybové hlásenie.

Poznámka: Ak je slovník, súbor default.policy a súbor default.auth pre systém modifikovaný, musíte reštartovať démony RADIUS spustením príkazu **stopsrc** a **startsrc** alebo použitím SMIT.

Súbor /etc/radius/proxy:

Súbor /etc/radius/proxy je konfiguračným súborom, ktorý podporuje funkciu proxy. Tento súbor mapuje známe sféry, do ktorých môže proxy server preposlať pakety.

Súbor /etc/radius/proxy používa IP adresu servera, ktorý pre ten-ktorý realm spracúva pakety a kľúč zdieľaný medzi týmito dvoma servermi.

Súbor obsahuje nasledujúce polia, ktoré môžete modifikovať pomocou SMIT:

- **Realm Name**
- **Next Hop IP address**
- **Shared Secret**

Nasleduje príklad súboru /etc/radius/proxy:

Poznámka:

Zdieľaný kľúč by mal mať dĺžku 16 znakov. Rovnaký zdieľaný kľúč musíte nakonfigurovať v nasledujúcom skoku servera RADIUS.

```
# @(#)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
```

Autentifikácia

Tradičná autentifikácia využíva meno a pevné heslo a vo všeobecnosti k nej dochádza vtedy, keď sa používateľ k počítaču prihlasuje po prvý raz alebo keď požaduje službu. RADIUS sa spolieha na autentifikačnú databázu pre ukladanie užívateľských ID, hesiel a iných informácií.

Pre autentifikáciu užívateľov môže server použiť lokálnu databázu, heslá systému UNIX alebo LDAP. Umiestnenie databázy sa konfiguruje v súbore servera /etc/radius/radiusd.conf počas nastavovania alebo aktualizáciou súboru prostredníctvom rozhrania SMIT. Bližšie informácie o konfiguračných súboroch protokolu RADIUS nájdete v časti “Konfiguračné súbory RADIUS” na strane 302.

Užívateľské databázy:

Softvér RADIUS môže používať rôzne databázy na uloženie užívateľských informácií.

Na ukladanie užívateľských informácií môžete použiť lokálnu databázu, databázu UNIX alebo LDAP.

UNIX:

Táto voľba autentifikácie na úrovni UNIX umožňuje protokolu RADIUS využívať na autentifikáciu používateľa metódu autentifikácie v lokálnom systéme.

Ak chcete používať lokálnu autentifikáciu systému UNIX, upravte pole **database_location** v súbore radiusd.conf, alebo v SMIT poli Database Location vyberte UNIX. Táto metóda autentifikácie volá za účelom autentifikácie ID používateľa a hesla rozhranie aplikačného programu (API) UNIX **authenticate()**. Heslá sa ukladajú v tom istom údajovom súbore, ktorý používa aj operačný systém UNIX, napríklad /etc/passwords. ID používateľov a heslá sa tvoria pomocou príkazu **mkuser** alebo prostredníctvom rozhrania SMIT.

Ak chcete používať databázu UNIX, v poli **Database Location** vyberte UNIX, ako to vidíte nižšie:

```
    Configure Server

RADIUS Directory            /etc/radius
*Database Location           [UNIX]
Local AVL Database File Name [dbdata.bin]
Local Accounting             [ON]

Debug Level                  [3]
.
.
.
```

Local:

Ak pole **database_location** súboru radiusd.conf, alebo ak SMIT položka Database Location obsahuje slovo Local, server RADIUS použije /etc/radius/dbdata.bin ako umiestnenie pre všetky užívateľské ID a heslá.

Lokálna databáza používateľov je paušálny súbor, ktorý obsahuje informácie o ID používateľov a heslách. Heslá sa ukladajú v hašovanom formáte. Hašovanie je technika rýchleho adresovania pre priame adresovanie údajov v priestore pamäte. Ak si želáte pridať, vymazať alebo modifikovať používateľské heslá, spustite príkaz **raddbm** alebo použite rozhranie SMIT. Keď sa spustí démon **radiusd**, prečíta súbor radiusd.conf a zavedie užívateľské ID a heslá do pamäte.

Poznámka: Maximálna dĺžka ID používateľa je 253 znakov a maximálna dĺžka hesla je 128 znakov.

Ak chcete používať lokálnu užívateľskú databázu, v poli **Database Location** vyberte Local, ako to vidíte v príklade nižšie:

```
    Configure Server

RADIUS Directory            /etc/radius
*Database Location           [Local]
Local AVL Database File Name [dbdata.bin]
Local Accounting             [ON]

Debug Level                  [3]
.
.
.
```

LDAP:

RADIUS môže použiť LDAP verziu 3 na ukladanie vzdialených užívateľských údajov.

RADIUS použije volania rozhrania API LDAP verziu 3 na vzdialený prístup k užívateľským údajom. Prístup LDAP Version 3 nastane, ak pole **database_location** v súbore /etc/radiusd.conf je nastavené na LDAP a ak je nakonfigurovaný názov servera, užívateľské ID administrátora LDAP a heslo administrátora LDAP.

AIX používa klientske knižnice LDAP Version 3, ktoré sú podporované a zbalené v IBM Tivoli Directory Server. LDAP je škálovateľný protokol a výhodou jeho používania je, že užívateľa a údaje v procese možno umiestniť do centralizovaného umiestnenia, čo uľahčuje správu servera RADIUS. Môžete použiť pomocný program príkazového riadka **ldapsearch** na prezeranie ľubovoľných údajov RADIUS.

Aj LDAP musí byť pred použitím pre RADIUS nakonfigurovaný a administrovaný.

Server RADIUS poskytuje súbory LDAP ldif na pridanie schémy RADIUS vrátane tried objektov a atribútov do adresára, ale LDAP musí byť nastavený a nakonfigurovaný.

Osobitná prípona sa vytvorí konkrétne pre RADIUS na použitie objektov LDAP RADIUS. Táto prípona je kontajner s názvom cn=aixradius a obsahuje dve objektové triedy, ako to popisuje “Konfigurácia servera RADIUS LDAP” na strane 313. Použijete súbor ldif dodaný RADIUS, ktorý vytvára príponu a schému RADIUS.

Keď ako autentifikačnú databázu použijete službu LDAP, máte k dispozícii nasledovné funkcie:

1. Užívateľská databáza, ktorú môžete vidieť a pristupovať na ňu zo všetkých serverov RADIUS
2. Zoznam aktívnych používateľov
3. Funkcia povolenia maximálneho počtu prihlásení na jedno ID používateľa
4. Typ protokolu **EAP**, ktorý možno konfigurovať zvlášť pre každého používateľa
5. Dátum ukončenia platnosti hesla

Ak chcete používať databázu LDAP, v poli **Database Location** vyberte LDAP, ako to vidíte na príklade nižšie:

```
    Configure Server

RADIUS Directory            /etc/radius
*Database Location           [LDAP]
Local AVL Database File Name [dbdata.bin]
Local Accounting             [ON]

Debug Level                  [3]
.
.
.
```

Súvisiace informácie:

IBM Directory Server

Konfigurácia servera RADIUS LDAP:

Keď sa konfiguruje autentifikácia používateľa služby LDAP, je potrebné aktualizovať serverovú schému LDAP. Administrátor systému LDAP musí pred definovaním užívateľov LDAP RADIUS pridať atribúty a triedy objektov definované serverom AIX RADIUS do adresára LDAP.

K serveru LDAP musíte pridať príponu. Prípona pre RADIUS má názov cn=aixradius. Prípona je charakteristický názov, ktorý identifikuje najvyššiu položku v hierarchii adresára.

Keď sa pridá prípona, adresár LDAP obsahuje prázdny kontajner. *Kontajner* je prázdnou položkou, ktorú je možné použiť na rozdelenie názvového priestoru. Kontajner je čosi podobné ako adresár súborového systému a pod ním sa môžu nachádzať adresárové položky. Informácie o profile používateľa možno teda do adresára LDAP pridávať prostredníctvom rozhrania SMIT. ID a heslo správcu LDAP sú uložené v súbore /etc/radius/radiusd.conf a možno ich konfigurovať na serveri RADIUS prostredníctvom rozhrania SMIT.

Za účelom organizácie informácií uložených v položkách adresára LDAP definuje schéma triedy objektov. Triedu objektu tvorí množina povinných a voliteľných atribútov. Atribúty majú podobu párov typ=hodnota, kde typ je definovaný jedinečným identifikátorom objektu (OID)a hodnota má definovanú syntax. Každá položka v adresári LDAP je inštanciou nejakého objektu.

Poznámka: Trieda objektu sama osebe ešte nedefinuje strom adresárových informácií alebo názvový priestor. Stane sa tak iba vtedy, keď sú vytvorené položky a špecifické inštancie tried objektov dostanú jedinečné charakteristické názvy. Keď napríklad trieda objektu kontajner dostane jedinečný charakteristický názov, môže byť asociovaná s dvoma ďalšími položkami, ktoré sú inštanciami triedy objektu organizačná jednotka. Výsledkom je stromová štruktúra alebo názvový priestor.

Triedy objektov sú pre server RADIUS špecifické a pri ich použití sa využíva súbor ldif. Niektoré z týchto atribútov sú existujúcimi atribútmi schémy LDAP a niektoré sú špecifické pre RADIUS. Nové triedy objektov RADIUS sú štrukturálne a abstraktné.

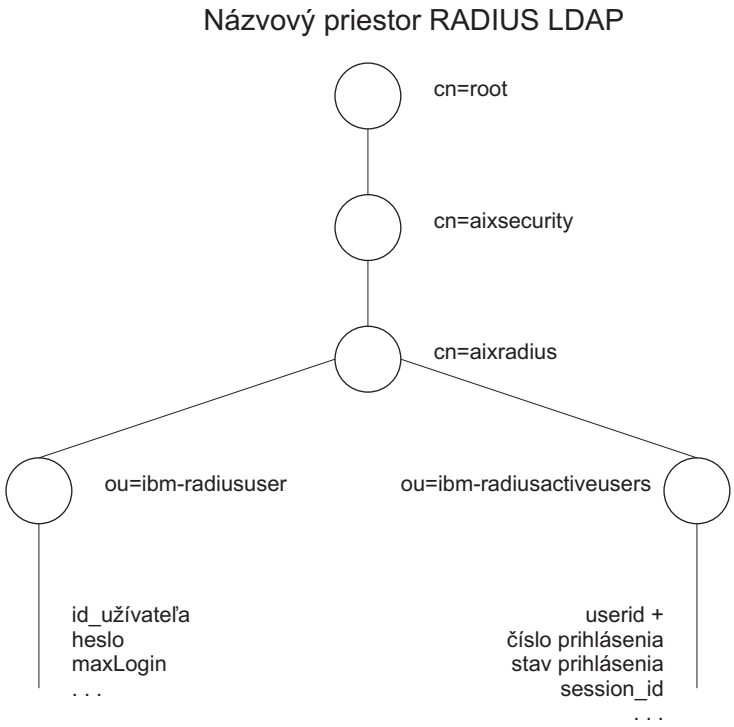
Z bezpečnostných dôvodov používajú väzby na server LDAP jednoduchú väzbu alebo volanie SASL API, **ldap_bind_s**, ktoré bude obsahovať charakteristický názov, CRAM-MD5 ako autentifikačnú metódu a heslo správcu LDAP. Týmto spôsobom sa po sieti prenášajú vlastne skôr stručné extrakty správ než samé heslá. CRAM-MD5 je bezpečnostný mechanizmus, ktorý nepotrebuje ani na jednej strane (klient resp. server) nijakú zvláštnu konfiguráciu.

Poznámka: Všetky atribúty v triedach objektov majú jedinú hodnotu.

Názvový priestor RADIUS LDAP:

Názvový priestor RADIUS LDAP má kontajner cn=aixradius v hornej časti svojej hierarchie. Pod kontajnerom cn=aixradius sú dve organizačné jednotky (OU). Tieto organizačné jednotky sú kontajnermi, ktoré pomáhajú tomu, aby boli položky jedinečné.

Nasledujúci obrázok je grafickým znázornením LDAP schémy služby RADIUS. Tento obrázok zobrazuje kontajnery a organizačné jednotky, ktoré sú znázornené krúžkami a pospájané čiarami alebo vetvami. Kontajner aixradius sa v strede rozvetvuje na dve organizačné jednotky: ibm-radiususer a ibm-radiusactiveusers. Pod kontajnerom ibm-radiususer sú zahrnuté kontajnery userid, password a maxLogin. Pod kontajnerom ibmradiusactiveusers sú zahrnuté kontajnery userid +, login number, login status a session_id. Nad kontajnerom aixradius sa nachádza kontajner aixsecurity a kontajner root je hore.



Obrázok 16. Názvový priestor RADIUS LDAP

Súbory schémy názvového priestoru LDAP:

Súbory schémy LDAP definujú triedy objektov a atribúty špecifické pre RADIUS pre názvový priestor LDAP.

Nasledujúce súbory schémy LDAP sú umiestnené v adresári /etc/radius/ldap:

IBM.V3.radiusbase.schema.ldif

Tento súbor definuje triedu objektu hornej úrovne pre server RADIUS (cn=aixradius). Súbor tiež vytvára nasledujúce vetvy pod triedou objektu cn=aixradius:

```
ou=ibm-radiususer
ou=ibm-radiusactiveusers
```

Požadované informácie môžete pridávať pomocou nasledovného príkazu:

```
ldapadd -D ldap_admin_id -w password -i /etc/radius/ldap/IBM.V3.radiusbase.schema.ldif
```

Tento príkaz môžete spustiť na systéme servera LDAP alebo ho môžete spustiť vzdialene s voľbou **-h** (názov hostiteľského systému).

IBM.V3.radius.schema.ldif

Tento súbor definuje triedy objektov a atribúty špecifické pre RADIUS.

Nové atribúty pre RADIUS a triedy objektov môžete pridávať zadaním nasledovného príkazu:

```
ldapmodify -D ldap_admin_id -w password -i  /etc/radius/ldap/IBM.V3.radius.schema.ldif
```

Ako umiestnenie databázy musíte prostredníctvom rozhrania SMIT zadať tiež LDAP a je potrebné zadať aj názov servera LDAP a heslo správcu. Potom, ako to vykonáte, môžete pridať užívateľov LDAP RADIUS do adresára prostredníctvom SMIT.

Trieda objektov užívateľských profilov:

Užívateľské profily LDAP musia byť zadané do systému predtým, než bude môcť server RADIUS autentifikovať užívateľa do systému. Profily obsahujú ID používateľa a heslo.

Objekty používateľských profilov poskytujú údaje o konkrétnych jednotlivcoch, ktorí majú prístup do siete, a obsahujú autentifikačné informácie. Na triedu objektov **ibm-radiusUserInstance** sa pristupuje synchrónne pomocou volaní API rozhraní LDAP z démona. Jedinečným poľom, ktoré tvorí začiatok charakteristického názvu, je ID používateľa. Pole **MaxLoginCount** ohraničuje počet možných prihlásení konkrétneho používateľa LDAP.

Trieda objektov zoznamov aktívnych prihlásení:

Zoznam aktívnych prihlásení LDAP predstavuje údaje obsahujúce informácie o momentálne prihlásených užívateľoch.

Existuje viacero záznamov na jedného užívateľa s počiatočným záznamom login_number = 1 až do čísla MaxLoginCount 5. ID relácie je zo správy RADIUS start_accounting. Čiastočne dokončené záznamy sa vytvoria, keď bude vytvorený objekt **ibm-radiusUserInstance**. Znamená to, že väčšina polí je prázdnych ešte pred prijatím paketov s údajmi o administrácii užívateľských kont RADIUS. Po prijatí správy RADIUS start_accounting sa objekt **ibm-radiusactiveusers** zaktualizuje a zadáva, že užívateľ je teraz práve prihlásený a jedinečné informácie o relácii budú zapísané do správneho čísla prihlásenia. Po prijatí správy stop_accounting sa informácia v zázname zoznamu aktívnych prihlásení vymaže. Evidencia aktívnych prihlásení sa aktualizuje, čiže teraz bude vykazovať, že dotyčný používateľ je v aktuálnej chvíli odhlásený. Čísla relácií v správach administrácie používateľských kont s informáciami "start" a "stop" sú tými istými jedinečnými číslami. Na triedu objektov sa bude pristupovať synchrónne pri volaniach API rozhrania LDAP.

Protokol Password authentication protocol:

Password Authentication Protocol (**PAP**) poskytuje bezpečnosť tak, že používateľské heslo zakóduje hašovacím algoritmom MD5 cez hodnotu, ktorú môžu navrhnúť ako klient tak aj server.

Funguje to nasledovne:

1. V paketoch ktoré majú užívateľské heslo, pole Authentication obsahuje 16 oktetové náhodné číslo, ktoré sa nazýva Request Authenticator.
2. Tento Autentifikačný reťazec požiadavky a zdieľaný kľúč klienta sa vložia do hašu MD5. Výsledkom je 16-oktetový haš.
3. Užívateľom zadané heslo sa blokovo rozmiestni do 16 oktetov s hodnotami null.
4. Haš z kroku 2 je XORed (Exclusive-OR) s doplneným heslom. Takto vyzerajúce údaje sa posielajú v pakete ako atribút *user_password*.
5. Server RADIUS vypočíta ten istý haš ako v kroku Step 2.
6. Týmto hašom je XORed s údajmi paketu z kroku Step 4, a tým sa obnovuje heslo.

Protokol Challenge handshake authentication protocol:

RADIUS podporuje pre ochranu hesiel aj používanie **CHAP** protokolu PPP.

Pri použití protokolu CHAP sa používateľské heslo neposiela cez sieť. Namiesto neho sa odošle haš MD5 hesla a server RADIUS zrekonštruuje tento haš z užívateľských informácií, vrátane uloženého hesla, potom ho porovná s hodnotou, ktorú odoslal klient.

Rozšíriteľný autentifikačný protokol:

Protokol Extensible Authentication Protocol (**EAP**) je protokol určený na podporu viacerých autentifikačných metód..

EAP špecifikuje štruktúru autentifikačnej komunikácie medzi klientom a autentifikačným serverom bez definovania obsahu autentifikačných údajov. Tento obsah definuje osobitná metóda **EAP**, ktorá sa používa na autentifikáciu. Medzi bežné metódy **EAP** patria:

- MD5-challenge
- One-time password
- Generic token card
- Transport layer security (TLS)

RADIUS využíva **EAP** tak, že zadáva atribúty RADIUS, ktoré sa použijú na prenos údajov **EAP** medzi serverom RADIUS jeho klientmi. Tieto údaje **EAP** môže potom server RADIUS odoslať priamo na back-end servery, ktoré rozličné autentifikačné metódy **EAP** vykonávajú.

Server AIX RADIUS podporuje len metódy EAP EAP-TLS a MD5-challenge.

Metódu EAP používanú na autentifikáciu užívateľa môžete nastaviť na úrovni užívateľa nastavením hodnoty v položke užívateľa v lokálnej databáze alebo LDAP.

Štandardne je protokol EAP pre každého užívateľa vypnutý.

Authorization

RADIUS umožňuje autorizáciu atribútov na užívateľa podľa definície v súboroch politiky autorizácie default.auth a default.policy.

Autorizačné atribúty sú platnými atribútmi protokolu RADIUS špecifikovanými v RFC a definovanými v súbore /etc/radius/dictionary. Autorizácia je nepovinná a závisí od toho, ako je nakonfigurovaný hardvér NAS alebo prístupový bod. Ak sa autorizačné atribúty vyžadujú, musíte ich nakonfigurovať. K autorizácii dochádza iba vtedy, ak prebehne úspešná autentifikácia.

Politiky sú konfigurovateľnými pármi atribútov používateľa a hodnôt, ktoré môžu riadiť spôsob prístupu daného používateľa k sieti. Politiky môžu byť definované ako globálne pre server RADIUS alebo špecifické pre užívateľa.

Dodávajú sa dva autorizačné konfiguračné súbory: /etc/radius/authorization/default.auth a default.policy. Súbor default.policy sa používa na párovanie prichádzajúcich paketov požiadaviek na prístup. Súbor obsahuje páry atribútov a hodnôt, ktoré sú najskôr prázdne a na požadované nastavenia je ich treba nakonfigurovať. Po autentifikácii politika rozhodne, či sa klientovi vráti paket udelenia prístupu alebo paket zamietnutia prístupu.

Okrem toho môže mať aj každý používateľ svoj vlastný súbor *user_id*.policy. Ak má používateľ svoj jedinečný súbor politiky vytvorený pre jeho konkrétne ID používateľa, potom sa najprv skontrolujú atribúty takéhoto súboru. Ak sa páry atribút-hodnota v súbore *user_id*.policy presne nezhodujú, potom sa skontroluje súbor default.policy. Ak sa dvojice atribútov v pakete požiadavky na prístup nezhodujú ani v jednom súbore, potom sa odošle paket zamietnutia prístupu. Ak sa nájde zhoda v jednom alebo v druhom súbore, klientovi bude odoslaný paket udelenia prístupu. Tak sa efektívne vytvoria dve úrovne politiky.

Súbor default.auth sa používa ako zoznam párov atribút-hodnota, ktoré sa majú vrátiť ku klientovi, akonáhle je skontrolovaná politika. Súbor default.auth obsahuje aj páry atribút-hodnota, ktoré sú spočiatku prázdne a a na požadované nastavenia je ich treba nakonfigurovať. Ak chcete nakonfigurovať požadované nastavenia autorizačných atribútov, musíte upraviť súbor default.auth alebo použiť rozhranie SMIT. Každý atribút, ktorý obsahuje hodnotu sa automaticky vráti do NAS v pakete udelenia prístupu.

Vytvorením súboru založeného na jedinečnom mene užívateľa s príponou .auth, napríklad ***user_id***.auth, môžete tiež definovať autorizačné atribúty vrátenia špecifické pre užívateľa. Tento voliteľný súbor musí byť umiestnený v adresári /etc/radius/authorization. Je tu panel SMIT, ktorý vám umožňuje vytvoriť a upraviť každý užívateľský súbor.

Všetky autorizačné atribúty užívateľa sa odošlú späť v pakete udelenia prístupu so všetkými predvolenými autorizačnými atribútmi, ktoré sa nachádzajú v súboroch default.auth alebo global.auth.

Ak sú hodnoty v súbore default.auth a v súbore *user_id*.auth spoločné, potom hodnoty používateľa majú prednosť pred predvolenými hodnotami. To platí pre niektoré globálne autorizačné atribúty (služby alebo prostriedky) pre všetkých používateľov a potom pre špecifickejšie úrovne autorizácie pre konkrétnych používateľov.

Poznámka: Na kombinovanie autorizačných atribútov s autorizačnými atribútmi špecifickými pre užívateľa sa namiesto súboru default.auth používa súbor global.auth, pokiaľ sa nevyžaduje iné správanie kombinácie.

Počnúc vydaním AIX, verzia 6.1, s technologickou úrovňou 6100-02, RADIUS podporuje autorizačný súbor global.auth. Tento súbor nahrádza a vylepšuje pôvodný zámer kombinovať autorizačné atribúty špecifické pre užívateľa (definované v súboroch user_id.auth) so sadou globálnych autorizačných atribútov.

Súbor user_id.auth nebude, na rozdiel od súboru default.auth, prepísaný podobnými atribútmi, ktoré boli nájdené v autorizačných súboroch špecifických pre užívateľa, ale namiesto toho sa s nimi skombinuje, čo umožní väčšiu flexibilitu pri udržiavaní autorizácií pre užívateľov.

Ak sú v súboroch default.auth a user_id.auth file atribúty spoločné, hodnoty užívateľa prepíšu predvolené hodnoty. Toto prepísanie predvolených hodnôt poskytuje niektoré predvolené autorizačné atribúty (služby alebo prostriedky) všetkým užívateľom a následne špecifickejšej úrovni autorizácie jednotlivých užívateľov.

To isté platí pre atribúty v súbore global.auth s výnimkou toho, že nebudú prepísané atribútmi user_id.auth. Namiesto toho budú skombinované atribúty v dvoch súboroch, čo je užitočné pri zadávaní atribútov špecifických pre predajcu (VSA).

Autorizačný proces prebieha takto:

1. V čase spustenia démona sa do pamäte načíta predvolená politika a autorizačné zoznamy zo súborov /etc/radius/authorization/default.policy, default.auth a default.auth.
2. Autentifikuj ID používateľa a heslo.
3. Skontrolujú sa páry atribútov a hodnôt v prichádzajúcom pakete.
   a. Skontroluj používateľský súbor ***user_id***.auth.
   b. Ak sa nenájde nijaká zhoda, skontroluj súbor default.policy.
   c. Ak sa nenájde žiadna zhoda, potom odošlite paket zamietnutia prístupu.
4. Uplatni autorizačné atribúty používateľa, ak nejaké existujú.
   a. Čítaj súbor /etc/radius/authorization/***user_id***.auth a súbor default.auth a obe položky porovnaj.
   b. Použite položku, ktorá je v súbore užívateľa nad predvolenou položkou.
   c. Skombinujte výsledné atribúty s atribútmi, ktoré sa nachádzajú v súbore global.auth.
5. Vráťte autorizačné atribúty v pakete udelenia prístupu.

Administrácia užívateľských kont

Server vykonávajúci administráciu užívateľských kont RADIUS zodpovedá za prijímanie požiadaviek na administráciu užívateľských kont od klienta a vrátenie odpovedí klientovi uvádzajúcich, že úspešne prijal požiadavku a zapísal údaje administrácie užívateľských kont.

Lokálnu administráciu užívateľských kont môžete zapnúť v súbore radiusd.conf.

Ak je klient nakonfigurovaný na použitie administrácie užívateľských kont RADIUS, na začiatku dodávky služby vygeneruje paket ACCOUNTING_START opisujúci typ dodávanej služby a užívateľa, ktorému je služba dodávaná. Klient pošle paket serveru vykonávajúcemu administráciu užívateľských kontRADIUS, ktorý vráti potvrdenie o prijatí paketu. Na konci poskytovania služby klient vygeneruje paket ACCOUNTING_STOP popisujúci typ služby, ktorá bola poskytnutá a (nepovinne) aj štatistiku ako napríklad uplynutý čas, vstupné a výstupné oktety alebo počty vstupných a výstupných paketov. Po prijatí paketu ACCOUNTING_STOP serverom vykonávajúcim administráciu užívateľských kont RADIUS, server vráti potvrdenie klientovi administrácie užívateľských kont o prijatí paketu.

Paket ACCOUNTING_REQUEST, či už ide o START alebo STOP, sa serveru RADIUS vykonávajúcemu administráciu používateľských kont odovzdáva prostredníctvom siete. Odporúča sa, aby sa klient pokúšal paket ACCOUNTING_REQUEST posielať až dovtedy, pokým nedostane potvrdenie o jeho doručení. V prípade, že primárny server je vypnutý alebo prostredníctvom danej konfigurácie proxy nedostupný, môže klient požiadavky preposielať aj na alternatívny server alebo alternatívne servery. Bližšie informácie o službách proxy nájdete v “Služby proxy” na strane 319.

Údaje administrácie užívateľských kont sa zapisujú v štandardnom formáte RADIUS *attribute=value* do lokálneho súboru /etc/var/radius/data/accounting. Zapísanými údajmi sú údaje administrácie používateľských kont v pakete opatrené časovou značkou. Ak evidenčný server RADIUS nedokáže úspešne zaznamenať evidenčný paket, neodošle potvrdenie **Accounting_Response** klientovi a do súboru syslog sa zaprotokolujú informácie o chybe.

Súbor /var/radius/data/accounting:

Súbor /var/radius/data/accounting zachytáva to, čo pošle klient v paketoch ACCOUNTING START a ACCOUNTING STOP.

Súbor /var/radius/data/accounting je po prvej inštalácii prázdny. Údaje sa do súboru zapíšu na základe toho, čo klient odošle v paketoch ACCOUNTING START a ACCOUNTING STOP.

Nasleduje vzor typu údajov, ktoré server AIX RADIUS zapíše do súboru /var/radius/data/accounting. Vaše informácie sa líšia v závislosti od nastavenia vášho systému.

Poznámka:

- Skontrolujte, či je súborový systém /var dostatočne veľký na to, aby mohol zvládnuť všetky údaje o administrácii užívateľských kont.
- Na analyzovanie údajov v tomto súbore možno použiť Perl skripty tretej strany. Príklady skriptov generujúcich správy z údajov o administrácii používateľských kont nájdete na http://www.pgregg.com/projects/radiusreport
- Evidenčné pakety sa môžu posielať aj cez proxy server.

```
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 was physically connected to port #1 on the hardware
 NAS-Port-Type = Async
 User-Name = "rod"
 Acct-Status-Type = Stop
 Acct-Authentic = RADIUS
 Service-Type = Framed-User
 Acct-Session-Id = "0000000C"   <-- note the session id's are the same so can match up start with stops
 Framed-Protocol = PPP
 Framed-IP-Address = 10.10.10.2   <-- IP address of user rod
 Acct-Terminate-Cause = User-Request  <-- user cancelled the session
 Acct-Input-Octets = 4016
 Acct-Output-Octets = 142
 Acct-Input-Packets = 35
 Acct-Output-Packets = 7
```

```
Acct-Session-Time = 120  <--- seconds
Acct-Delay-Time = 0
Timestamp = 1085687119  <--- note "rod" was only logged on for 120 seconds (2 minutes)
```

Služby proxy

Služby proxy umožňujú serveru RADIUS preposielať požiadavky z NAS ďalšiemu serveru RADIUS a potom vrátiť NAS správu s odpoveďou. Služby proxy vychádzajú z názvu realmu.

Server RADIUS sa môže chovať naraz ako proxy server aj ako back-end server. Tento mechanizmus sa vzťahuje rovnako na pakety administrácie používateľských kont ako aj na autentifikačné pakety. Služba proxy je v súbore radiusd.conf štandardne zakázaná.

Sféry:

Sféry sú identifikátory umiestnené pred a za hodnotami bežne sa nachádzajúcimi v atribúte User-Name, ktoré môže server RADIUS použiť na identifikáciu servera na kontaktovanie s cieľom spustiť autentifikáciu a proces administrácie užívateľských kont.

Nasledujúci príklad znázorňuje použitie sfér s RADIUS:

Používateľ, napríklad *Jozef*, je zamestnancom spoločnosti *XYZ* v Bratislave. Realmom pre túto oblasť je *BRA*. *Jozef* je však v danej chvíli na dlhšej pracovnej ceste v Žiline. Realm pre Žilinu je *ZIA*. Keď sa teda *Jozef* spája s realmom *ZIA*, odovzdaný atribút User-Name bude mať tvar *BRA/Jozef*. Server sféry *ZIA* RADIUS dostane oznam, že tento paket musí byť preposlaný serveru, ktorý vykonáva autentifikáciu a administráciu užívateľských kont pre užívateľov sféry *BRA*.

Atribút user-name sféry:

Pakety autentifikácie a administrácie užívateľských kont sú smerované cez sféru založenú na atribúte **User-Name**. Tento atribút definuje poradie sfér, cez ktoré paket prechádza, s cieľom nasmerovať ho do konečného servera, ktorý vykonáva autentifikáciu alebo administráciu užívateľských kont.

Pakety sú smerované zreťazením sfér dohromady v atribúte **User-Name**. Skutočné realmy, ktoré sa vkladajú do atribútu **User-Name**, ktorý s konečnou platnosťou určuje cestu paketu, sú ponechané na rozhodnutie administrátora, ktorý rozmiestňuje usporiadanie pre RADIUS. Názvy skokov realmov sa môžu umiestniť pred atribút **User-Name** a rovnako sa môžu umiestniť aj zaň. Najznámejšie znaky pre vymedzenie rôznych realmov sú lomka (/), ako vymedzovač predpony pred atribútom **User-Name** a ampersand (&), ako vymedzovač prípony za atribútom **User-Name**. Popisovacie značky sa konfigurujú v súbore radiusd.conf. Syntaktická analýza atribútu **User-Name** sa vykonáva zľava doprava.

Toto je príklad atribútu **User-Name**, ktorý používa iba metódu predpôn:

```
USA/TEXAS/AUSTIN/joe
```

Toto je príklad atribútu **User-Name**, ktorý používa iba metódu prípon:

```
joe@USA@TEXAS@AUSTIN
```

Použiť možno aj obidve metódy, predponu aj príponu zároveň. Keď sa zadávajú skoky medzi realmami, ktorými bude paket prechádzať, je dôležité mať na pamäti, že poradie skokov sa analyzuje (číta) zľava doprava a všetky skoky definované predponami sa spracúvajú skôr ako skoky definované príponami. Autentifikácia užívateľa alebo zápis evidenčných údajov musí byť vykonaný na jednom uzle.

Nasledujúci príklad, v ktorom sú použité obidve metódy, prináša rovnaký výsledok ako predchádzajúce príklady:

```
USA/joe@TEXAS@AUSTIN
```

Konfigurácia proxy služieb:

Konfiguračné informácie RADIUS proxy sa nachádzajú v súbore proxy v adresári /etc/radius.

Úvodný súbor proxy obsahuje vzory položiek. V súbore proxy sú tri polia: **Realm Name**, **Next Hop IP address** a **Shared Secret**.

Ak chcete nakonfigurovať pravidlá proxy, vyberte z nasledujúcich::

```
    Configure Proxy Rules

List all Proxy
Add a Proxy
Change / Show Characteristics of a Proxy
Remove a Proxy
```

Vyberte voľbu **List all Proxy** pre čítanie súboru /etc/radius/proxy a zobrazenie troch polí v stĺpcovom formáte. Takto vyzerajú záhlavia stĺpcov:

```
realm_name   next_hop_address  shared_secret
```

Vyberte **Add a Proxy**, aby sa zobrazila nasledujúca obrazovka. Informácie sa načítavajú z panela a údaje sa pripájajú na koniec súboru /etc/radius/proxy.

Každý skok v reťazi proxy používa zdieľaný kľúč medzi dvoma servermi RADIUS. Zdieľaný kľúč sa nachádza v /etc/radius/proxy_file. Zdieľaný kľúč by mal byť jedinečný pre každý jeden skok proxy v reťazi.

Bližšie informácie o vytváraní zdieľaných kľúčov nájdete v “Súbor /etc/radius/clients” na strane 308.

Ak chcete pridať proxy, vyberte si polia ukázané nižšie:

```
            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)
```

Voľba príkazu **Change/Show** vypíše na obrazovku zoznam názvov realmov. Tento zoznam sa zobrazí na vysúvacom displeji, z ktorého si musíte názov realmu vybrať.

Voľba **Remove a Proxy** vypíše zoznam názvov realmov. sa zobrazí na vysúvacom displeji, z ktorého si používateľ musí názov realmu vybrať. Po voľbe názvu - a ešte než dôjde k vlastnému odstráneniu realmu - sa zobrazí overovací vysúvací displej.

Nasledujúci príklad je časťou konfiguračných informácií proxy zo súboru radiusd.conf:

```
#-------------------------------------------------------------------#
#       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
```

Konfigurácia servera RADIUS:

Démon servera RADIUS používa niekoľko konfiguračných súborov. Informácie o konfigurácii servera sú uložené v súbore /etc/radius/radiusd.conf. Zbalený konfiguračný súbor servera sa dodáva so štandardnými hodnotami.

Poznámka: Nasleduje vzorový SMIT panel RADIUS Configure Server:

```
    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         []
```

Pomocné programy protokolovania

Server RADIUS používa SYSLOG na protokolovanie informácií o činnosti a chybách.

Existujú tri úrovne protokolových informácií:

0 Protokolujú sa len problémy alebo chyby a spúšťanie démonov.

3 Protokoluje kontrolný záznam správ access_accept, access_reject*, discard a error.

Poznámka: Správy discard sú zaprotokolované, keď je prichádzajúci paket neplatný a paket odpovedí nie je vygenerovaný.

9 Obsahuje protokolové informácie úrovne 0 a 3 a mnohé ďalšie. Spúšťa sa iba protokolovanie úrovne 9 za účelom ladenia.

Predvolenou úrovňou protokolovania je úroveň 3. Protokolovanie na úrovni 3 sa používa na zvýšenie úrovne auditovania servera RADIUS. V závislosti od toho, na akej úrovni server protokoluje, môžete aktivity uložené v protokole využívať na overovanie podozrivých vzorov aktivity. Ak dôjde k narušeniu bezpečnosti, výstup programu SYSLOG možno použiť na zistenie spôsobu a času, kedy k narušeniu došlo a prípadne aj na určenie miery, do akej sa niekomu podarilo taký prístup získať. Tieto informácie sú užitočné pre vývoj lepších zabezpečovacích opatrení, aby sa zamedzilo problémom v budúcnosti.

Súvisiace informácie:

Konfigurácia RADIUS na použitie démona syslogd:

Ak chcete použiť SYSLOG na prezeranie chybových informácií a informácií o aktivitách, musíte zapnúť démona syslogd.

Ak chcete zapnúť démona syslogd, postupujte nasledovne.

1. Upravte súbor /etc/syslog.conf na pridanie položky: local4.debug var/adm/ipsec.log. Na zaznamenanie prenosov a udalostí protokolov IP Security použite prostriedok *local4*. Používajú sa štandardné úrovne priorít operačného systému. Ak je prenos cez tunelové prepojenia a filtre protokolov IP Security stabilný a správny, mali by ste použiť úroveň priority *debug*.

   Poznámka: Protokolovanie udalostí filtra môže vytvoriť dôležitú aktivitu na hostiteľovi IP Security a spotrebovať veľký rozsah pamäte.
2. Uložte /etc/syslog.conf file.
3. Prejdite do adresára zadaného pre protokolový súbor a vytvorte prázdny súbor s rovnakým názvom. Vo vyššie uvedenom prípade by ste vykonali zmenu na adresár /var/adm a spustili príkaz **touch** nasledovne:
   ```
   touch ipsec.log
   ```
4. Spustite príkaz **refresh** pre podsystém syslogd nasledovne:
   ```
   refresh -s syslogd
   ```

Konfigurácia nastavenia výstupu SYSLOG:

V súbore radiusd.conf môžete nastaviť Debug_Level 0, 3 alebo 9 v závislosti od toho, koľko informácií o ladení chcete mať zahrnutých vo výstupe SYSLOG.

Predvoleným nastavením je 3. Časť ladenia súboru radiusd.conf je podobná tomuto:

```
#.
#.
#.
#   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]  #
#                                                                 #
#-----------------------------------------------------------------#
```

Nasledujúce príklady zobrazujú vzorový výstup pre rôzne úrovne ladenia.

Paket administrácie užívateľských kont s úrovňou ladenia 3

```
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() **
```

Pakety administrácie užívateľských kont na úrovni 9

```
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.poli
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
```

Paket autentifikácie úrovne 0

```
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
```

Paket autentifikácie úrovne 3

```
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() **
```

Paket autentifikácie úrovne 9

```
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.poli
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() **
```

Uplynutie doby platnosti hesla

Uplynutie doby platnosti hesla umožňuje notifikáciu klienta RADIUS keď skončí platnosť užívateľského hesla a aktualizáciu užívateľského hesla prostredníctvom protokolu RADIUS.

Funkcia uplynutia platnosti hesla vyžaduje podporu štyroch ďalších typov paketov a jeden nový atribút. Nové typy paketov sa dodávajú so slovníkom AIX a funkcia uplynutia platnosti hesla musí byť zapnutá.

Povolenie obnovy hesla s uplynutou platnosťou cez RADIUS nemusí byť žiaduce v každej inštalácii služby RADIUS. Položka radiusd.conf v súbore vám poskytuje možnosť povoliť alebo zakázať podporu zmeny hesla s uplynutou platnosťou cez RADIUS. Štandardne je táto možnosť zakázaná. Môžete pridať správu užívateľskej odpovede Password_Expired_Reply_Message a tá sa vráti v pakete Password-Expired. Atribúty hesiel, starého ako aj nového, sa musia šifrovať a dešifrovať metódou PAP.

Špecifické atribúty dodávateľa

Špecifické atribúty dodávateľa (VSA) definujú dodávatelia serverov vzdialeného prístupu (zvyčajne sú to dodávatelia hardvéru), aby prispôsobili fungovanie servera RADIUS na ich serveroch.

Špecifické atribúty predajcu sú nevyhnutné vtedy, ak chcete používateľom povoliť viac ako jeden typ prístupu. Atribúty VSA sa môžu použiť v kombinácii s atribútmi, ktoré definoval server RADIUS.

Atribúty VSA sú voliteľné, ale ak si hardvér NAS pre konfiguráciu vyžaduje ďalšie atribúty, aby mohol správne fungovať, musíte pridať atribúty VSA do slovníkového súboru.

Špecifické atribúty predajcu možno tiež využiť na ďalšiu autorizáciu. VSA môžete použiť na autorizáciu spolu s atribútmi **User-Name** a **Password**. Na strane servera obsahuje súbor politiky autorizácie používateľov zoznam atribútov, ktoré sa majú v pakete Access-Request od toho-ktorého používateľa kontrolovať. Ak paket neobsahuje atribúty, ktoré sú uvedené v súbore užívateľov, potom sa do NAS odošle access_reject. Atribúty VSA sa môžu použiť aj ako zoznam dvojíc atribút=hodnota v súbore ***user_id***.policy.

Toto je ukážka časti slovníka so špecifickými atribútmi predajcu:

```
########################################################################
#                                                                      #
#   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
```

Podpora správ pre odpoveď RADIUS

Správa pre odpoveď je text, ktorý vytvoríte a nakonfigurujete v súbore radiusd.conf.

Je určený pre zariadenie NAS alebo prístupový bod, ktoré ho vrátia používateľovi v podobe reťazca. Môže sa jednať o správu typu úspech, zlyhanie alebo námietka. Sú to čitateľné textové polia a ich obsahy závisia od implementácie a konfigurujú sa pri konfigurácii servera. Štandardne sú tieto atribúty nastavené tak, že neobsahujú nijaký text. Môžete nakonfigurovať všetky, žiaden alebo jeden, dva alebo tri atribúty.

RADIUS podporuje nasledujúce atribúty správ pre odpoveď:

- Accept Reply-Message
- Reject Reply-Message
- CHAP Reply-Message
- Password Expired Reply-Message

Tieto atribúty sa pridávajú do konfiguračného súboru radiusd.conf a pri spúšťaní démona sa načítavajú do štruktúry globálnej konfigurácie. Tieto hodnoty sa nastavujú pomocou panelov SMIT RADIUS v rámci voľby **Konfigurácia servera**. Každý reťazec môže mať maximálne 256 bajtov.

Funkcia sa implementuje nasledovne:

1. Keď sa spustí démon **radiusd**, prečíta súbor radiusd.conf a nastaví atribúty Reply-Message.
2. Keď bude prijatý paket žiadosti o prístup, vykoná sa autentifikácia užívateľa.
3. Ak odpoveď autentifikácie bude prístup povolený, potom sa skontrolujte text Accept Reply-Message. Ak sa tam text nachádza, reťazec sa vráti v pakete udelenia prístupu.
4. Ak bude autentifikácia zamietnutá, potom sa označí text Reject Reply-Message, ktorý bude vrátený v pakete zamietnutia prístupu.
5. Ak bude autentifikácia odopretá, potom sa označí atribút CHAP Reply-Message, ktorý sa odošle ako súčasť paketu Access-Challenge.

Konfigurácia IP oblasti servera RADIUS

Pomocou servera RADIUS môžete dynamicky priraďovať IP adresy z oblasti IP adries.

Alokácia IP adries je súčasťou procesu autorizácie a vykonáva sa po autentifikácii. Správca systému musí každému užívateľovi priradiť jedinečné IP. Ak chcete IP adresu poskytnúť užívateľovi dynamicky, server RADIUS ponúka tri voľby:

- Atribút Framed Pool
- Použitie špecifického atribútu dodávateľa
- Zhromažďovanie IP na strane servera RADIUS

Atribút Framed Pool

IP oblasť *poolname* musí byť definovaná v NAS (Network Access Server). NAS musí vyhovovať RFC2869, aby mohol server RADIUS odoslať atribút **Framed-Pool** v balíku Access-Accept (atribút typu 88). Správca systému musí pre užívateľa nakonfigurovať NAS a aktualizovať autorizačné atribúty tak, že atribút **Framed-Pool** zahrnie buď do globálneho súboru default.auth alebo do súboru user.auth v serveri RADIUS. Slovníkový súbor na serveri RADIUS bude obsahovať tento atribút:

```
ATTRIBUTE    Framed-Pool    88       string
```

Ak NAS nedokáže používať viaceré adresové oblasti, NAS bude tento atribút ignorovať. Adresová oblasť v NAS obsahuje zoznam IP adries. NAS dynamicky vyberie niektorú IP adresu, ktorá je definovaná v špecifikovanej oblasti a priradí ju užívateľovi.

Špecifické atribúty dodávateľa

Niektorí nezávislí predajcovia softvéru (ISV) nemôžu používať atribút **Framed-Pool**, ale dokážu definovať oblasti IP adries. Server RADIUS dokáže tieto adresové oblasti využiť s použitím modelu špecifických atribútov dodávateľa (VSA). Napríklad, Cisco NAS poskytuje atribút s názvom Cisco-AVPair. Slovníkový súbor na serveri RADIUS bude obsahovať tento atribút:

```
VENDOR     Cisco           9
ATTRIBUTE  Cisco-AVPair    1       string
```

Keď NAS odošle paket Access-Request, bude v ňom zahrnutý tento atribút s Cisco-AVPair=”ip:addr-pool=*poolname*”, pričom *poolname* je názov adresovej oblasti, ktorá je definovaná v NAS. Po autentifikácii a autorizácii požiadavky vráti server RADIUS tento atribút v pakete Access-Accept. NAS potom môže definovanú oblasť používať na alokáciu IP adresy užívateľovi. Správca systému musí pre užívateľa nakonfigurovať NAS a aktualizovať autorizačné atribúty tak, že atribút VSA zahrnie buď do globálneho súboru default.auth alebo do súboru user.auth v serveri RADIUS.

Zhromažďovanie IP na strane servera Radius

Server RADIUS sa dá nakonfigurovať, aby vygeneroval IP adresu z oblasti IP adries. IP adresa sa vráti v atribúte Framed-IP-Address paketu Access-Accept.

Správca systému môže oblasť IP adries definovať pomocou rozhrania SMIT. Adresy sa uchovávajú v súbore /etc/radius/ippool_def. *Poolnames* sú definované v súbore etc/radius/clients. Správca systému musí nakonfigurovať aj číslo pre NAS-Port. Démon servera RADIUS používa informácie zo súborov etc/radius/clients a /etc/radius/ippool_def na vytvorenie údajových súborov. Akonáhle sa démon spustí, správca systému nebude môcť zmeniť alebo pridať *poolnames* alebo rozsahy IP adries, kým sa servery RADIUS nezastavia. Keď sa spustí démon servera RADIUS, prečíta konfiguračný súbor (/etc/radius/radius.conf) a ak je povolená IP alokácia (Enable_IP_Pooling=YES), príznak globálnej IP alokácie (IP_pool_flag) nastaví na hodnotu On. Démon potom zisťuje, či existuje súbor poolname.data. Ak súbor existuje, prečíta ho a tieto informácie si uchová v zdieľanej pamäti. Súbor a zdieľanú pamäť bude následne aktualizovať na základe požiadaviek, ktoré prichádzajú od klientov. Ak súbor neexistuje, potom démon vytvorí nový súbor a použije na to informácie zo súborov etc/radius/clients a /etc/radius/ippool_def. Limit maximálnej veľkosti súboru poolname.data je 256 MB (Limit veľkosti segmentu AIX). Ak bude mať súbor poolname.data viac ako 256 MB, server RADIUS zaprotokoluje chybové hlásenie a ukončí sa.

Démon získa podrobnosti o IP oblasti zo súboru /etc/radius/ippool_def a tabuľku IP adries pre každý názov oblasti uchová v zdieľanej pamäti. Tabuľka má položky pre NAS-IP-address, NAS-port a príznak IN USE. Démon uchováva hašovaciu tabuľku, ktorá je zakľúčovaná podľa NAS-IP NAS-port. Keď požiadavky prichádzajú od viacerých užívateľov, UDP zaradí požiadavky do frontu a démon obnoví údaje NAS-IP a NAS-port z požiadavky. S použitím týchto informácií preverí, či bol pre NAS definovaný *poolname*, to znamená že skontroluje informácie načítané zo súboru etc/radius/clients.

Démon sa pokúsi z oblasti získať nepoužívanú adresu. Ak bude nepoužívaná adresa k dispozícii, pomocou príznakov NAS-IP a NAS-port sa označí ako “in use” a vráti sa do servera RADIUS. Démon vloží IP adresu do atribútu **Framed-IP-Address** a do NAS sa vráti v pakete akceptovania. Súbor poolname.data bude tiež aktualizovaný, aby bol zosynchronizovaný s informáciami v zdieľanej pamäti.

Ak oblasť neexistuje, alebo ak existuje, ale už nemá žiadne nepoužívané adresy, do servera RADIUS sa vráti chyba. Chyba Could not allocate IP address sa zaprotokoluje do protokolového súboru a server RADIUS odošle do NAS paket Access-Reject.

Kódy chyby sú:

- NOT_POOLED – Pre **nas_ip** nie je definovaná žiadna oblasť.
- POOL_EXHAUSTED – Pre **nas_ip** je oblasť definovaná, ale všetky adresy z oblasti sa v súčasnosti používajú.

Keď požiadavka na autentifikáciu pochádza z kombinácie NAS a NAS-port, ktorá už má alokovanú IP adresu, démon vráti do oblasti predchádzajúcu alokáciu, pričom príznak IN USE označí hodnotou Off a v tabuľke vyčistí položky NAS-IP-address a NAS-port. Následne bude v oblasti alokovať novú IP adresu.

IP adresa sa vráti do oblasti aj vtedy, keď server RADIUS prijme z NAS paket Accounting-Stop. Paket Accounting-Stop musí obsahovať položky NAS-IP-address a NAS-port. Démon pristúpi na súbor ippool_mem v nasledujúcich prípadoch:

- Prichádza požiadavka na získanie novej IP adresy. Nastaví príznak IN USE na hodnotu True.
- Bude prijatý paket Accounting-Stop. Uvoľní IP adresu tak, že príznak “in use” nastaví na hodnotu False.

V obidvoch prípadoch volania systému zdieľanej pamäte zabezpečia, že údaje v zdieľanej pamäti a v súboroch poolname.data budú synchrónne. Správca systému môže IP alokáciu nastaviť na hodnotu ON alebo OFF pomocou parametra *Enable_IP_Pooling* v konfiguračnom súbore servera RADIUS (radiusd.conf). Je to užitočné v prípadoch, kedy má správca systému priradenú IP adresu buď v globálnom súbore default.auth alebo v súbore user.auth. Ak chce správca systému túto pridelenú IP adresu používať, musí nastaviť Enable_IP_Pool = NO.

Príklad súboru /etc/radius/ippool_def, ktorý bol vytvorený prostredníctvom SMIT:

Pool Name	Start Range	End Range
Floor5	192.165.1.1	192.165.1.125
Floor6	192.165.1.200	192.165.1.253

Nasleduje príklad súboru /etc/radiusclients, ktorý bol vytvorený prostredníctvom SMIT:

NAS-IP	Shared Secret	Pool Name
1.2.3.4	Secret1	Floor5
1.2.3.5	Secret2	Floor6
1.2.3.6	Secret3	Floor5
1.2.3.7	Secret4	

V príklade, vyššie, pre NAS-IP-Address 1.2.3.7, je názov oblasti prázdny. V tomto prípade sa zhromažďovanie IP pre tento NAS nevykoná (aj keby globálny IP_pool_flag = True). Keď príde paket Access-Request, server RADIUS vykoná autentifikáciu a autorizáciu. Ak bude úspešná, odošle statickú IP adresu, ktorá je definovaná v požiadavke alebo v globálnom súbore default.auth alebo v súbore user.auth, v pakete Access-Accept. V tomto prípade sa atribút NAS-Port nevyžaduje.

Ak má zhromažďovanie IP hodnotu True, správca systému aj statickú IP adresu definoval ako súčasť globálneho súboru default.auth alebo súboru user.auth alebo ako súčasť paketu Access-Request. Server RADIUS nahradí túto IP adresu IP adresou, ktorá bolo alokovaná v názve definovanej oblasti pre tento NAS. Ak sa všetky IP adresy v oblasti používajú, server zaprotokoluje chybu (oblasť je plná) a odošle paket Access-Reject. Server bude ignorovať každú statickú IP adresu, ktorá je definovaná v súboroch auth.

Ak má zhromažďovanie IP hodnotu True a pre NAS je definovaný platný názov oblasti, keď paket Access-Request príde z NAS-IP a nemá definovaný NAS-Port, server odošle paket Access-Reject.

Nasleduje príklad súboru Floor5.data, ktorý vytvoril démon:

IP Address	NAS-IP	NAS-Port	In Use
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

Nasleduje príklad súboru Floor6.data, ktorý vytvoril démon:

IP Address	NAS-IP	NAS-Port	In Use
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

Keď je potrebné uvoľniť všetky alokované IP adresy pre špecifikovaný NAS (napríklad, keď sa NAS zastaví), možno bude nutné uvoľniť všetky IP adresy zo všetkých oblastí, aby sa dal inicializovať súbor *poolname*.data. Správca systému to môže vykonať použitím SMIT s nasledujúcimi akciami ponuky:

- Clear IP Pool for a Client
- Clear entire IP Pool

Panely SMIT pre IP oblasť

V Client Configuration vyberte **Add a Client**, môžete zadať voliteľné **Pool Name**. Názov môže mať maximálne 64 znakov. Keď je **Pool Name** prázdne, zhromažďovanie IP sa nevykoná a server RADIUS priradí IP adresu, ktorú definoval správca systému prostredníctvom autorizačného atribútu **Framed-IP-Address**.

Keď je vybratá **IP Pool**, zobrazia sa nasledujúce voľby:

- List all IP Pools
- Create an IP Pool
- Change/Show Characteristics of an IP Pool
- Delete an IP Pool
- Clear IP Pool for a Client
- Clear entire IP Pool

List all IP Pools: Túto voľbu použite na vypísanie zoznamu polí **Pool Name**, **Start Range IP address** a **Stop Range IP address**.

Create an IP Pool: Túto voľbu použite na pridanie názvu oblasti, začiatku rozsahu a konca rozsahu. Tieto údaje sa pripoja na koniec súboru ippool_def. Kontroly sa vykonávajú, aby sa zaručilo, že neexistujú žiadne duplicitné názvy oblastí a že rozsahy IP adries sú nesúvislé. Táto akcia sa môže vykonať iba vtedy, ak nie sú spustení démoni servera RADIUS.

Change/Show Characteristics of an IP Pool: Táto voľba zobrazí zoznam názvov oblastí v rozbaľovacom paneli. V tomto paneli musíte vybrať špecifický názov oblasti. Keď vyberiete názov oblasti, zobrazí sa panel s vybratým názvom.

Keď stlačíte kláves Enter, údaje pre tento názov oblasti sa aktualizujú do súboru ippool_def. Táto akcia sa môže vykonať iba vtedy, ak nie sú spustení démoni servera RADIUS.

Delete an IP Pool: Výber tejto voľby zobrazí zoznam názvov oblastí, ktoré môžete vybrať. Keď vyberiete názov oblasti, zobrazí sa panel **Are You Sure** v ktorom potvrdíte vymazanie vybratého názvu oblasti. Skript **rmippool** bude vyvolaný, aby zo súboru ippool_def vymazal názov vybratej oblasti. Táto akcia sa môže vykonať iba vtedy, ak nie sú spustení démoni servera RADIUS.

Clear IP Pool for a Client: Táto voľba označí položku **IN-USE** pre IP adresy, ktoré patria do NAS, hodnotou 0, čo znamená, že odteraz budú k dispozícii všetky IP adresy pre tento NAS. Táto akcia sa môže vykonať iba vtedy, ak nie sú spustení démoni servera RADIUS.

Clear Entire IP Pool: Keď vyberiete túto voľbu, zobrazí sa kontextový panel **Are You Sure**, v ktorom potvrdíte, že sa má vyčistiť celý súbor ippool_mem. Táto akcia sa môže vykonať iba vtedy, ak nie sú spustení démoni servera RADIUS.

SMIT panely pre RADIUS

Keď na konfiguráciu servera RADIUS používate SMIT, polia označené hviezdičkou (*) sú povinné polia.

Rýchla cesta pre SMIT je:

```
smitty radius
```

RADIUS má nasledovnú hlavnú ponuku:

```
       RADIUS Server

Configure Server
Configure Clients
Configure Users
Configure Proxy Rules
Advanced Server Configuration
Start RADIUS Server daemons
Stop RADIUS Server daemons
```

Nasledujúci snímok obrazovky ukazuje vzorový SMIT panel Configure Server pre RADIUS:

```
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)        []
```

Podrobné informácie pomoci SMIT sú k dispozícii pre všetky polia a voľby ponúk po stlačení klávesu **F1**.

Generátor náhodných čísel

Náhodné čísla sa vyžadujú pri generovaní poľa Authenticator v pakete RADIUS.

Je dôležité zabezpečiť čo možno najlepší generátor, pretože narušiteľ by sa mohol pokúsiť ľsťou primäť server RADIUS, aby odpovedal na predvídanú požiadavku, a potom by odpoveď použil, aby sa pri budúcich žiadostiach o prístup falošne vydával za tento server RADIUS. AIX RADIUS Server používa na generovanie pseudonáhodných čísel rozšírenie jadra **/dev/urandom**. Toto rozšírenie jadra zbiera prostredníctvom pseudoovládača zariadenia entropické vzorky z hardvérových zdrojov. Toto zariadenie prešlo testovaním NIST, aby sa zaručila správna náhodnosť.

Podpora globalizácie

Príkaz RADIUS **raddbm** a panely SMIT podporujú globalizáciu a používajú štandardné volania rozhrania API globalizácie systému AIX s cieľom poskytovať túto funkciu.

Súvisiace informácie

Príkazy: **installp**,**mkuser** a **raddbm**

Ochrana pred neoprávneným vniknutím v systéme AIX

Ochrana pred neoprávneným vniknutím v systéme AIX zisťuje nevhodné, neoprávnené alebo iné údaje, ktoré by mohli byť považované za škodlivé pre systém.

Nasledujúca časť popisuje rôzne typy zisťovania neoprávneného vniknutia, ktoré umožňuje operačný systém AIX.

Súvisiace informácie

Príkazy: **chfilt**, **ckfilt**, **expfilt**, **genfilt**, **impfilt**, **lsfilt**, **mkfilt**, **mvfilt**, **rmfilt**.

Zisťovanie neoprávneného vniknutia

Zisťovanie neoprávneného vniknutia je činnosťou, pri ktorej sa monitorujú a analyzujú systémové udalosti za účelom zachytenia a odmietnutia akéhokoľvek pokusu o neoprávnený vstup do systému. V AIX sa zisťovanie neoprávneného prístupu alebo pokusu o neoprávnený prístup vykonáva sledovaním určitých akcií a následným použitím pravidiel filtrovania pre tieto akcie.

Poznámka: Ak chcete detekciu prienikov povoliť, musíte si na hostiteľský systém nainštalovať sady súborov bos.net.ipsec. Tieto technológie detekcie sú založené na existujúcich bezpečnostných komponentoch AIX Internet Protocol Security (IPsec).

Pravidlá filtrovania používajúce porovnávanie so vzorom:

Porovnávanie so vzorom je metóda, ktorú používajú pravidlá filtrovania IPsec na filtrovanie sieťových paketov. Filtrovacím vzorom môže byť textový reťazec, hexadecimálny reťazec alebo súbor obsahujúci viac ako jeden vzor. Po vytvorení vzorového pravidla filtrovania a potom, ako bude tento vzor zistený v tele niektorého sieťového paketu, výsledkom bude preddefinovaná akcia pravidla filtrovania.

Pravidlá filtrovania pomocou porovnávania so vzorom sa používajú iba na preverovanie prichádzajúcich sieťových paketov. Ak si želáte do tabuľky filtrovacích pravidiel pridať pravidlo filtrovania, použite príkaz **genfilt**. Pravidlá filtrovania vygenerované pomocou tohto príkazu sa nazývajú manuálnymi filtrovacími pravidlami. Na aktivovanie alebo deaktivovanie pravidiel filtrovania používajte príkaz **mkfilt**. Príkaz **mkfilt** možno použiť aj na ovládanie funkcie protokolovania filtrovania.

Vzorový súbor môže obsahovať zoznam textových alebo hexadecimálnych vzorov (v každom riadku zoznamu jeden vzor). Pravidlá filtrovania pomocou porovnávania so vzorom možno využiť aj na ochranu proti vírusom, preplňovaniu vyrovnávacej pamäte a ďalším rizikám hroziacim zo sieťovej prevádzky.

Pravidlá filtrovania pomocou porovnávania so vzorom môžu mať negatívny dopad na výkonnosť systému, ak sa používajú v priveľkej miere a ak sa v nich uplatňuje priveľký počet vzorov. Najlepšie je udržiavať pole ich použitia čo najužšie. Ak sa napríklad má vzor známeho vírusu použiť na filtrovanie **sendmail**, potom vo filtrovacom pravidle zadajte ako cieľový port SMTP pre **sendmail** 25. Takto umožníte prechod všetkej ostatnej sieťovej prevádzky bez toho, aby porovnávanie vzorov spôsobovalo spomaľovanie systému.

Príkaz **genfilt** rozpoznáva formát vzorov používaných v niektorých verziách ClamAV a dokáže ho pochopiť.

Súvisiace informácie:

Príkaz genfilt

Príkaz mkfilt

Webová lokalita ClamAV

Typy vzorov:

Existujú tri základné typy vzorov: textové, hexadecimálne a súborové. Pravidlá filtrovania používajúce porovnávanie so vzorom sa používajú len pre prichádzajúce pakety.

Textový vzor

Vzorom textového filtra je reťazec ASCII, ktorý sa podobá tomuto:

```
GET /../../../../../../../../
```

Hexadecimálny vzor

Hexadecimálny vzor sa podobá tomuto:

```
0x33c0b805e0cd16b807e0cd1650558becc7460200f05d0733ffb8c800b9fffff3abb00150
e670e47132c0e67158fec03c8075f033c033c9b002fa99cd26fb4183f90575f5c3
```

Poznámka: Hexadecimálny vzor sa od textového vzoru odlišuje úvodnými znakmi 0x.

Súbory obsahujúce textové vzory

Súbor môže obsahovať zoznam textových alebo hexadecimálnych vzorov (v každom riadku zoznamu jeden vzor). Ukážkové vzorové súbory si môžete pozrieť na adrese http://www.clamav.net.

Pravidlá filtrovania vylučujúce hostiteľa a port:

Nastavením vylučovacieho pravidla filtrovania môžete ovplyvniť prístup vzdialeného hostiteľa alebo páru vzdialený hostiteľ a port na lokálny počítač.

Vylučovacie pravidlo filtrovania dynamicky vytvára účinné pravidlo, ktoré po splnení pravidlom zadaných kritérií odmietne prístup vzdialeného hostiteľa alebo páru vzdialený hostiteľ a port na lokálny počítač.

Keďže bežná prax je taká, že útoku po sieti predchádza skenovanie portov, pravidlá filtrovania vylučujúce porty sa pri predchádzaní prienikom obzvlášť dobre uplatňujú tak, že detekujú tento druh agresívneho správania.

Napríklad, ak lokálny hostiteľ nepoužíva serverový port 37, čo je časový server, potom by sa vzdialený hostiteľ nemal pokúšať o prístup na port 37 - ibaže by spustil skenovanie portov. Aplikujete teda na port 37 vylučovacie pravidlo filtrovania. Ak sa potom vzdialený hostiteľ pokúsi o prístup na tento port, vylučovacie pravidlo filtrovania vytvorí účinné pravidlo, ktoré tomuto hostiteľovi zablokuje ďalší prístup na daný port, a to na čas, ktorý do tohto pravidla zadáte (pole **platnosť do**).

Ak je pole **expiration time** vylučovacieho pravidla nastavené na 0, dynamicky vytvorenému účinnému vylučovaciemu pravidlu sa platnosť neskončí.

Poznámka:

1. Doba uplynutia platnosti zadaná portovým vylučovacím filtrovacím pravidlom sa vzťahuje iba na toto dynamicky vytvárané efektívne pravidlo.
2. Dynamicky vytvárané efektívne pravidlá si možno prezerať iba prostredníctvom príkazu **lsfilt -a**.

Pravidlá filtrovania vylučujúce hostiteľa

Po splnení kritérií pravidla filtrovania vylučujúceho hostiteľa dynamicky vytvorené účinné pravidlo zablokuje alebo vylúči celú prevádzku siete zo vzdialeného hostiteľa po uvedenú dobu uplynutia platnosti.

Pravidlá filtrovania vylučujúce port

Po splnení kritérií pravidla filtrovania vylučujúceho port dynamicky vytvorené účinné pravidlo len zablokuje alebo vylúči celú prevádzku siete z daného portu vzdialeného hostiteľa až do uplynutia doby exspirácie.

Pravidlá filtra zachovávajúceho stav:

Filtre zachovávajúce stav preverujú informácie ako napríklad zdrojové a cieľové adresy, čísla portov a stav. Potom, uplatnením filtrovacích pravidiel IF, ELSE alebo ENDIF na tieto príznaky hlavičiek môžu systémy zachovávajúce stav samy prijímať rozhodnutia o filtrovaní, a to skôr v kontexte celej relácie než v kontexte jednotlivých paketov a ich hlavičkových informácií.

Kontrola zameraná na zachovávanie stavu skúma prichádzajúce a odchádzajúce komunikačné pakety. Keď sú pravidlá filtrovania zachovávajúceho stav aktivované prostredníctvom príkazu **mkfilt -u**, pravidlá v bloku ELSE sa preverujú vždy, až pokým nie je nie sú splnené podmienky pravidla IF. Keď je splnená podmienka či pravidlo IF, pravidlá v bloku IF sa používajú až dovtedy, pokým nie sú príkazom **mkfilt -u** reaktivované pravidlá filtrovania.

Príkaz **ckfilt** skontroluje syntax pravidiel filtrovania zachovávajúceho stav a prehľadným spôsobom ich zobrazí na displeji - napríklad takto:

```
%ckfilt -v4
Beginning of IPv4 filter rules.
Rule 2
IF Rule 3
    IF Rule 4
        Rule 5
    ELSE Rule 6
        Rule 7
    ENDIF Rule 8
ELSE Rule 9
    Rule 10
ENDIF Rule 11
Rule 0
```

Načasované pravidlá:

Načasované pravidlá udávajú dĺžku času (v sekundách), po ktorý sa má pravidlo filtrovania používať po svojom efektívnom uplatnení príkazom **mkfilt -v [4|6] -u**.

Čas uplynutia platnosti pravidla sa zadáva príkazom **genfilt -e**. Bližšie informácie nájdete pri vysvetlivkách k príkazom **mkfilt** a **genfilt**.

Poznámka: Časovač nemá nijaký vplyv na pravidlá IF, ELSE alebo ENDIF. Ak sa čas uplynutia platnosti zadá do pravidla vylúčenia hostiteľa alebo portu, bude sa tento čas vzťahovať iba na efektívne pravidlo vytvorené daným vylučovacím pravidlom. Vylučovacie pravidlá nemajú nijaký čas uplynutia platnosti.

Prístup k pravidlám filtrovania zo SMIT

Pravidlá môžete nakonfigurovať zo SMIT.

Pri konfigurácii pravidiel filtrovania zo SMIT postupujte takto:

1. Z príkazového riadka zadajte príkaz: smitty ipsec4
2. Vyberte si **Advanced IP Security Configuration**.
3. Vyberte si **Configure IP Security Filter Rules**.
4. Vyberte si **Add an IP Security Filter Rule**.

```
                        Pridajte pravidlo filtrovania bezpečnosti IP

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
```

Voľby pre pole action sú: permit, deny, shun_host, shun_port, if, else, endif.

Ak je zadaný vzorový súbor, potom tento súbor musí byť po aktivovaní filtrovacích pravidiel príkazom **mkfilt -a** čitateľný. Pravidlá filtrovania sú uložené v databáze /etc/security/ipsec_filter.

AIX Security Expert

AIX Security Expert poskytuje centrum pre všetky nastavenia bezpečnosti (TCP, NET, IPSEC, systém a auditovanie).

AIX Security Expert je nástroj vylepšenia systémovej bezpečnosti. Je súčasťou sady súborov **bos.aixpert**. AIX Security Expert poskytuje nastavenia jednoduchých ponúk pre vysokú, strednú a nízku a štandardnú úroveň bezpečnosti AIX, ktoré integrujú vyše 300 nastavení konfigurácie bezpečnosti a zároveň poskytujú kontrolu všetkých elementov bezpečnosti pre skúsených administrátorov. AIX Security Expert možno použiť na implementáciu vhodnej úrovne bezpečnosti bez nutnosti čítať veľké množstvo dokumentov o vylepšení systému a individuálne implementovať každý element bezpečnosti.

AIX Security Expert môžete použiť na vykonanie snímky konfigurácie bezpečnosti. Túto snímku môžete použiť na nastavenie rovnakej konfigurácie bezpečnosti na iných systémoch. Toto vám ušetrí čas a zaistí, že všetky systémy v podnikovom prostredí budú mať správnu konfiguráciu bezpečnosti.

AIX Security Expert môžete spustiť z nástroja SMIT alebo môžete použiť príkaz **aixpert**.

Nastavenia AIX Security Expert

K dispozícii sú tieto primárne nastavenia bezpečnosti:

Vysoká úroveň bezpečnosti
Vysoká úroveň bezpečnosti

Stredná úroveň bezpečnosti
Stredná úroveň bezpečnosti

Nízka úroveň bezpečnosti
Nízka úroveň bezpečnosti

Rozšírená bezpečnosť
Užívateľom nastavená bezpečnosť

Štandardné nastavenia AIX
Pôvodné nastavenie bezpečnosti v systéme

Undo Security
NIektoré AIX Security Expertkonfiguračné nastavenia je možné vrátiť späť

Check Security
Poskytuje podrobné hlásenie o aktuálnych nastaveniach bezpečnosti

Vylepšenie bezpečnosti AIX Security Expert

Vylepšenie bezpečnosti chráni všetky elementy systému sprísnením bezpečnosti alebo implementáciou vyššej úrovne bezpečnosti.

Vylepšenie bezpečnosti pomáha zabezpečiť adekvátnosť a vhodnosť všetkých rozhodnutí a nastavení konfigurácie bezpečnosti. Aby ste zlepšili bezpečnosť systému AIX, možno budete musieť zmeniť stovky nastavení konfigurácie bezpečnosti.

AIX Security Expert poskytuje ponuku na centralizáciu efektívnych spoločných nastavení konfigurácie bezpečnosti. Tieto nastavenia sú založené na rozsiahlom výskume vhodného zabezpečenia systémov UNIX. K dispozícii sú predvolené nastavenia bezpečnosti pre široké potreby prostredia bezpečnosti (vysoká, stredná a nízka úroveň bezpečnosti) a skúsení administrátori môžu nastaviť každé nastavenie konfigurácie bezpečnosti nezávisle.

Konfigurácia systému na príliš vysokú úroveň bezpečnosti môže spôsobiť odmietnutie potrebných služieb. Napríklad **telnet** a **rlogin** sú pre vysokú úroveň bezpečnosti vypnuté, pretože heslo prihlásenia sa zasiela po sieti nezašifrované. Ak je systém nakonfigurovaný na príliš nízku úroveň bezpečnosti, je zraniteľný voči bezpečnostným hrozbám. Keďže každý podnik má svoju vlastnú jedinečnú sadu bezpečnostných požiadaviek, preddefinované nastavenia konfigurácie vysokej, strednej a nízkej úrovne bezpečnosti sú ako východisko pre konfiguráciu bezpečnosti vhodnejšie než presná zhoda bezpečnostných požiadaviek daného podniku.

Praktický prístup k používaniu AIX Security Expert je nastaviť testovací systém (v reálnom testovacom prostredí), podobný produkčnému prostrediu, v ktorom bude umiestnený. Nainštalujte potrebné podnikové aplikácie a spustite AIX Security Expert cez GUI. AIX Security Expert zanalyzuje tento bežiaci systém v dôveryhodnom stave. V závislosti od bezpečnostných volieb, ktoré ste vybrali, AIX Security Expert zapne ochranu skenovania portov a auditovanie, blokuje sieťové porty, ktoré nie sú používané podnikovými aplikáciami alebo inými službami, ako aj mnohé iné nastavenia bezpečnosti. Po opätovnom testovaní s nastavenými týmito bezpečnostnými nastaveniami, bude systém pripravený na nasadenie do produkčného prostredia. Taktiež, môžete jednoducho použiť súbor XML AIX Security Expert, ktorý definuje bezpečnostnú politiku alebo konfiguráciu tohto systému, na implementáciu rovnakej konfigurácie na podobné systémy vo vašom podniku.

Bližšie informácie o vylepšovaní bezpečnosti nájdete v publikácii NIST Special Publication 800-70, NIST Security Configurations Checklist Program for IT Products.

Secure by default

Štandardné zabezpečenie - Secure By Default (SbD) predstavuje koncept inštalácie minimálnej sady softvéru v bezpečnej konfigurácii.

Voľba inštalácie Secure by Default (SbD) v AIX nainštaluje odľahčenú verziu TCP klienta a sady súborov servera bez citlivých príkazov a súborov. Sady súborov **bos.net.tcp.client** a **bos.net.tcp.server** sú súčasťou inštalácie SbD a

obsahujú všetky príkazy a súbory okrem aplikácií, ktoré umožňujú prenos hesiel cez sieť v čistom textovom formáte, ako sú **telnet** a **ftp**. Navyše sú aplikácie, ktoré je možné použiť (napríklad **rsh**, **rcp** a **sendmail**), vyňaté zo sád súborov SbD.

Záverečný automatizovaný proces inštalácie SbD zavádza konfiguračné nastavenia AIX Security Expert na vysokej úrovni zabezpečenia. Toto môžete urobiť spustením príkazu **aixpert** zo skriptu /etc/firstboot: /usr/sbin/aixpert -f /etc/security/aixpert/core/SbD.xml -p 2>/etc/security/aixpert/log/firstboot.log

Počítač môžete vyňať z režimu SbD zmenou premennej ODM *SbD_STATE* na sbd_disable, opätovným nainštalovaním sady súborov **bos.net.tcp.client** a **bos.net.tcp.server** a použitím AIX Security Expert na návrat systému do jeho predvolenej úrovne bezpečnosti.

Štandardne zabezpečený SbD systém nie je možné nainštalovať s použitím metódy inštalácie so zachovaním ani s migráciou súborov. SbD predstavuje samostatnú inštalačnú cestu.

Poznámka: Keď pomocou servisného balíka aktualizujete systém, ktorý je v režime SbD, po aktualizácii verzie nebude aktualizovaný systém v režime SbD.

Bezpečne nakonfigurovaný systém môžete dosiahnuť aj bez použitia inštalačnej voľby SbD. Napríklad bezpečnostné voľby AIX Security Expert High, Medium alebo Low je možné nakonfigurovať počas bežnej inštalácie.

Rozdiely medzi systémom inštalovaným pomocou SbD a bežnou inštaláciou s konfiguráciou AIX Security Expert High Level Security sú najlepšie vykreslené preskúmaním príkazu **telnet**. V oboch prípadoch je príkaz **telnet** vypnutý. V inštalácii typu SbD sa binárny súbor alebo aplikácia **telnet** na systém nikdy ani nenainštaluje.

Keď použijete inštaláciu SbD, nasledujúce služby sa buď nenainštalujú do systému počas inštalácie, alebo budú zakázané. Ak nie sú v systéme nainštalované niektoré z týchto služieb, nie je možné v systéme pristupovať k týmto príkazom alebo ich spúšťať. V prípade, že tieto príkazy alebo programy budete potrebovať, nepoužite inštalačnú voľbu SbD. Rovnako, keď nejaké skripty, vzdialené programy alebo závislé sady súborov vyžadujú niektorý z týchto príkazov či programov, nepoužite voľbu inštalácie SbD.

Služba	**Program**	**Argumenty**
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

Služba	Program	Argumenty
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

Taktiež, niektoré funkcie softvéru IBM Systems Director Console for AIX vrátane portletu HealthMetrics nie sú k dispozícii, keď operačný systém AIX spustíte v režime SbD. Tieto funkcie môžete povoliť inštaláciou sád súborov vyžadovaných na spustenie danej funkcie.

Šírenie bezpečnostnej politiky cez LDAP

LDAP môžete použiť na distribúciu XML konfiguračných súborov AIX Security Expert. AIX Security Expert môžete použiť na kopírovanie konfigurácie bezpečnosti z jedného systému do druhého. Vďaka tomu môžu mať podobné systémy rovnakú konfiguráciu bezpečnosti. Takáto konzistentnosť pomáha redukovať zraniteľnosť zabezpečenia.

Odporúčaný postup je pomocou AIX Security Expert nakonfigurovať jeden systém a nastaviť úroveň zabezpečenia v súlade s podnikovou bezpečnostnou stratégiou a s prostredím, v ktorom bude tento systém fungovať. Táto konfigurácia je uložená v súbore /etc/security/aixpert/core/appliedaixpert.xml. Súbor môžete potom presunúť na nakonfigurovaný a dôveryhodný LDAP server. Ďalšie systémy s pripojením k tomuto serveru LDAP automaticky objavia tento konfiguračný súbor pomocou príkazu **aixpertldap**.

Každý existujúci server LDAP môže byť aktualizovaný so schémou aixpert, aby boli konfiguračné súbory XML pre aixpert prenesené na každého pripojeného klienta. Ak na serveri LDAP nie je aktualizovaná schéma aixpert, aktualizujte schému aixpert na serveri LDAP zadaním nasledujúceho príkazu: ldapmodify -c -D <bindDN> -w <bindPwd> -i /etc/security/ldap/sec.ldif. Keď je server LDAP aktualizovaný s touto schémou aixpert, klienti môžu umiestniť ich konfiguračné súbory XML na server LDAP pomocou voľby -u pre príkaz **aixpertldap**. Tieto konfiguračné súbory sa musia aktualizovať manuálne.

Poznámka: Táto funkcia je založená na modeli dôveryhodnosti LDAP. Užívatelia, ktorí majú oprávnenie zapisovať na LDAP, môžu upraviť údaje prenesené užívateľmi iných počítačov. Podobne, ak klient LDAP obsahuje bezpečnostné chyby, môžete zistiť stav zabezpečenia iných klientov LDAP v konfiguračných súboroch XML pre AIX Security Expert pre tohto klienta.

Napríklad súbor appliedaixpert.xml môže byť uložený na LDAP serveri pod názvom **BranchOfficeSecurityProfile**. Inak nakonfigurovaný súbor appliedaixpert.xml môže byť uložený napríklad pod názvom **InternetDirectAttachedSystemsProfile**. Keď pomocou programu AIX Security Expert konfigurujete ďalšie systémy pripojené cez protokol LDAP, tieto bezpečnostné profily sú automaticky ponúknuté ako voľby ponuky. Administrátor systému tak môže vybrať bezpečnostný profil, ktorý najlepšie vyhovuje príslušnému prostrediu a firemnej bezpečnostnej politike.

Potom je použitý AIX Security Expert na zabezpečenie systému. Úplný zoznam konfiguračných nastavení zabezpečenia, implementovaných v systéme, je uložený v súbore /etc/security/aixpert/core/appliedaixpert.xml. Tento súbor je bezpečnostná politika pre tento systém. Táto bezpečnostná politika je porovnaná, keď je použitá voľba AIX Security Expert Check Security. Túto bezpečnostnú politiku môžete tiež skopírovať a použiť na iné systémy, vďaka čomu vytvoríte konzistentné zabezpečenie všetkých systémov vo vašom prostredí. Bezpečnostnú politiku môžete skopírovať na iné systémy dvomi spôsobmi: manuálne a cez LDAP.

Kópia bezpečnostnej politiky AIX Security Expert

AIX Security Expert môžete použiť na kopírovanie bezpečnostných politík z jedného systému na iný.

AIX Security Expert môžete spustiť na jednom systéme, pričom bezpečnostnú politiku použijete aj na iné systémy. Bob chce napríklad použiť AIX Security Expert na jeho šiestich systémoch AIX. Bob použije nastavenia bezpečnosti na

jednom systéme (alfa) s nastaveniami vysokej, strednej, nízkej, rozšírenej alebo štandardnej bezpečnosti AIX. Otestuje kompatibilitu tohto systému v svojom prostredí a ak mu tieto nastavenia vyhovujú, tie isté nastavenia môže použiť na ostatných systémoch AIX podľa názvu. Skopíruje nastavenia zo systému alfa do systému, na ktorom chce použiť rovnaké nastavenia bezpečnosti, skopírovaním súboru /etc/security/aixpert/core/appliedaixpert.xml zo systému alfa na iný systém.

Poznámka: Nevytvárajte kópiu tohto súboru na iných systémoch v tom istom adresári pod tým istým názvom súboru, pretože počas implementácie bezpečnostnej politiky príkaz **aixpert** prepíše súbor /etc/security/aixpert/core/appliedaixpert.xml.

Radšej skopírujte bezpečnostnú politiku systému alfa do adresára /etc/security/aixpert/custom/. Tak bude ďalší systém schopný nájsť a uplatniť bezpečnostnú politiku systému alfa cez grafické užívateľské rozhranie na riadenie systému AIX Security Expert, alebo priamo pomocou príkazu **aixpert**.

Napríklad, ak súbor bezpečnostnej politiky appliedaixpert.xml systému alfa bol umiestnený na iné systémy ako /etc/security/aixpert/custom/*PolitikaAlfa*, príkaz aixpert -f /etc/security/aixpert/custom/*PolitikaAlfa* okamžite použije túto bezpečnostnú politiku a tento systém bude mať bezpečnostnú konfiguráciu identickú so systémom alfa. Navyše, ak je politika systému alfa v tomto adresári, je viditeľná a môže byť použitá cez konzolu riadenia na iných systémoch, pomocou Aix Security Expert -> Overview and Tasks -> Customized Options -> AlphaPolicy.

Prispôsobiteľné politiky zabezpečenia s užívateľom definovanými XML pravidlami AIX Security Expert

Na konfiguráciu jedinečných bezpečnostných politík môžete použiť súbory vo formáte XML.

AIX Security Expert dynamicky rozpozná tieto súbory XML. Všetky vytvorené súbory bezpečnostných politík XML by mali byť uložené v adresári /etc/security/aixpert/custom/ so súborom popisu. Preto, keď k AIX Security Expert pristupujete cez grafické rozhranie konzoly, naplno využijete bohaté funkcie XML v DTD aixpert.

DTD vyzerá takto:

```
<?xml version='1.0'?>

<!--START-->

<!ELEMENT AIXPertSecurityHardening (AIXPertEntry+)>

<!-- AIXPertEntry by mala obsahovať iba jednu inštanciu nasledujúcich elementov. -->

<!ELEMENT AIXPertEntry (AIXPertRuleType,
  AIXPertDescription, AIXPertPrereqList, AIXPertCommand,
  AIXPertArgs,AIXPertGroup)>

<!-- AIXPertEntry by mala mať jedinečný názov. -->

<!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)*>
```

Názov AIXPertEntry je jedinečný v rámci súboru XML. Tento názov bude určovať grafické tlačidlo, keď je tento súbor zobrazený cez systémovú konzolu na ceste Aix Security Expert -> Overview and Tasks -> Customized Options -> <xml file=""></xml>.

<!ELEMENT AIXPertRuleType EMPTY>
Súbor XML by mal byť špecifikovaný ako voliteľný.

<!ATTLIST AIXPertRuleType type (LLS|MLS|HLS|DLS|SCBPS|Prereq|Custom) "DLS"
Súbor XML by mal byť špecifikovaný ako voliteľný.

<!ELEMENT AIXPertDescription (#PCDATA)>
Keď je zobrazený cez uvedené grafické rozhranie, text popisu je zobrazený ako vyskakovacie okno, keď je nad týmto tlačidlom kurzor myši.

<!ELEMENT AIXPertPrereqList (#PCDATA)>
Pre toto pravidlo môžete určiť vyžadované pravidlo. Vyžadované pravidlo musí vrátiť hodnotu 0, predtým než sa aixpert pokúsi implementovať toto pravidlo. Ak je súbor XML zobrazený v grafickom rozhraní, toto pravidlo bude zobrazené ako nedostupné, ak nebolo vyhovené vyžadovanému pravidlu. Ak vytvárate vyžadované pravidlo, typ AIXPertRuleType musí byť 'Prereq'.

Pole AIXPertDescription vyžadovaného pravidla by malo popisovať, čo treba vykonať, aby bolo splnené vyžadované pravidlo. Ak je pole Custom rules nedostupné, pretože nebolo vyhovené niektorému z vyžadovaných pravidiel, užívateľovi je zobrazené okno popisu vyžadovaného pravidla, v ktorom je popísané, aké kroky sú potrebné na vyhovenie podmienkam tohto pravidla.

<!ELEMENT AIXPertCommand (#PCDATA)>
Tento element musí uvádzať úplnú cestu a príkaz, ktorý aixpert vykoná pre toto bezpečnostné pravidlo, napr. /usr/bin/ls.

<!ELEMENT AIXPertArgs (#PCDATA)*>
Tento element musí uvádzať argumenty pre tento príkaz, napr. -l

<!ELEMENT AIXPertGroup (#PCDATA)*>
Keď je množina pravidiel aixpert zobrazená v grafickom rozhraní, je možné ich zoskupiť. Napríklad, množina pravidiel v skupine pravidiel aixpert môže byť spoločne "Bezpečnosť siete".

Prísna kontrola slabých hesiel

Táto funkcia v AIX kontroluje a vyhľadáva slabé heslá, keď sa heslá menia. Keď je táto voľba vybratá s AIX Security Expert, táto dodatočná kontrola prebehne, keď si užívateľ vyberá alebo mení heslo. Kontrola bráni použitiu bežných anglických slov a 1000 najbežnejších krstných mien používaných v USA.

Kontrolné ciele COBIT, ktoré podporuje AIX Security Expert

AIX Security Expert podporuje okrem nastavení bezpečnosti na vysokú, strednú, nízku, predvolenú AIX a rozšírenú aj úroveň zabezpečenia SOB-COBIT Best Practices Security.

Kongres Spojených štátov uzákonil právnu normu 'Sarbanes-Oxley Act of 2002' na ochranu investorov zvýšením presnosti a spoľahlivosti firmami zverejňovaných finančných informácií. Funkcia riadenia cieľov COBIT pomáha administrátorom systémov pri konfigurácii, správe a audite IT systémov, ktoré musia byť v súlade s touto právnou normou. Nástroj SOX Configuration Assistant môžete spustiť prostredníctvom príkazového riadka aixpert. Táto funkcia pomáha so splnením požiadaviek SOX definovaným v paragrafe 404 zákona Sarbanes-Oxley Act, ale nástroj AIX Security Expert SOX Configuration Assistant automaticky implementuje bezpečnostné nastavenia, ktoré sa bežne spájajú s odporúčanými postupmi COBIT pre interné kontroly SOX, paragraf 404. AIX Security Expert ďalej poskytuje funkciu auditu SOX, ktorá hlási audítorovi, či je systém aktuálne nakonfigurovaný týmto spôsobom. Táto funkcia umožňuje automatizáciu konfigurácie systému v súlade s IT SOX a taktiež automatizáciu procesu samotného auditu.

Keďže SOX neposkytuje informácie o tom, ako musí byť v súlade s časťou 404, IT priemysel sa riadi pravidlami uvedenými na webovej stránke by www.isaca.org/. Sú to pravidlá Control Objectives for Information and related Technology (COBIT).

AIX Security Expert podporuje nasledujúce kontrolné ciele:

- Vynútenie politiky hesiel
- Hlásenia o narušení a aktivite v súvislosti s bezpečnosťou

- Prevencia, zisťovanie a náprava škodlivého softvéru a neoprávneného softvéru
- Architektúra firewallu a pripojenia k verejným sieťam

AIX Security Expert nepodporuje všetky atribúty uvedené pod jednotlivými kontrolnými cieľmi. Podporované atribúty a ich príslušné kontrolné ciele sú zhrnuté v nasledujúcich tabuľkách:

Vynútenie politiky hesiel

Popis	Nastavenia bezpečnosti
Maximálny vek hesla	maxage=13
História vynútenia hesiel	histsize=20
Minimálny vek hesla	minage=1
Minimálna dĺžka hesla	minlen=8
Obsahuje najmenej 6 znakov	Minalpha=6
Podobnosť s minulým heslom	mindiff=4
Počet dní varovania o uplynutí doby platnosti hesla	pwdwarntime=14

Hlásenie o narušení a aktivite v súvislosti s bezpečnosťou

Popis	Nastavenia bezpečnosti	Poznámky
Auditovanie povolené	áno	
Bez priameho prihlasovania koreňového užívateľa	áno	
Povoliť auditovanie pre eskaláciu privilégií	áno	AIXpert využíva auditovaciu udalosť USER_SU. Uistite sa, že je táto udalosť zapnutá.

Zisťovanie a náprava škodlivého softvéru

AIX Security Expert sa opiera o funkciu spúšťania dôveryhodného softvéru v systéme AIX, ktorá zaručuje, že so softvérom nikto nemanipuloval. Príkaz **trustchk** kontroluje konzistentnosť objektov zaregistrovaných v databáze Trusted Software.

Nastavenie firewalu

AIX Security Expert zapne IPSec a povolí filtračné pravidlá, aby sa vyhol skúmaniu portov. Porty, ktorým sa vyhýba, sú uvedené v nasledujúcej tabuľke:

Služba	Popis
Tcp/11, udp/11	Systat
Tcp/13, udp/13	Daytime
(RFC 867) Tcp/19, udp/19	Generátor znakov
Tcp/25	Simple Mail Transfer (SMTP)
Tcp/43, udp/43	Who Is (prezývka)
Tcp/63, udp/63	Whois++
Tcp/67, udp/67	Bootstrap protokolový server (bootps)
Tcp/68, udp/68	Bootstrap protokolový klient (bootpc)
Tcp/69, udp/69	Jednoduchý prenos súborov

Služba	Popis
(tftp) Tcp/79, udp/79	Finger
Tcp/87	Súkromná terminálová linka
Tcp/110	Protokol Post office protocol – verzia 3 (POP3)
Udp/111	SUN Remote Procedure Call
Tcp/113	Autentifikačná služba (auth)
Udp/123	Sieťový časový protokol (NTP)
Udp/161	SNMP
Udp/162	SNMPTRAP
Tcp/194	Protokol IRC (Internet Relay chat)
Tcp/443	Protokol http cez TLS/SSL
Tcp/511	PassGo
Tcp/514	Cmd (shell)
Tcp/520	Názvový server pre rozšírené súbory server (efs)
Tcp/540	Uucpd (uucp)
Tcp/546	DHCPv6 Client
Tcp/547	DHCPv6 Server
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	Vyhradený

Použitie kontrolných cieľov COBIT pomocou produktu AIX Security Expert

Pomocou príkazu **aixpert –l s** môžete použiť úroveň SCBPS na systém. Príslušný auditovací protokol vygenerujete zapnutím udalosti **AIXpert_apply**. Všetky zlyhania (či počas vykonávania predbežných krokov alebo počas použitia) budú nahlásené do **stderr** a auditovacieho podsystému, ak je povolený.

Kontrola súladu so SOX-COBIT, audit a funkcia predbežného auditu

Pomocou príkazu **aixpert –c –l s** môžete skontrolovať súlad systému so štandardmi SOX-COBIT. AIX Security Expert overuje iba súlad s podporovanými kontrolnými cieľmi. Akékoľvek porušenia zistené počas kontroly budú nahlásené. Štandardne sa správy o narušeniach posielajú do **stderr**.

Ten istý príkaz (**aixpert –c –l s**) môžete použiť aj na vygenerovanie správy o audite súladu so SOX-COBIT. Aby to bolo možné, nastavte a povoľte auditovací podsystém. Uistite sa, že auditovacia udalosť **AIXpert_check** je zapnutá. Po nastavení auditovacieho podsystému znovu spustite príkaz **aixpert –c –l s**. Tento príkaz vygeneruje auditovací protokol pre každý neúspešný kontrolný cieľ. Pole **Status** v protokole auditu bude označené ako **neúspešné**. Protokol tiež obsahuje príčinu zlyhania, ktorú môžete zobraziť použitím voľby **-v** v príkaze **auditpr**.

Keď k príkazu **aixpert –c –l s** pridáte voľbu **–p**, v hlásení o audite budú uvedené aj úspešné kontrolné ciele. Tieto položky protokolu budú mať v poli so stavom uvedené **Ok**.

Môžete použiť aj príkaz **aixpert -c -l s -p** na vygenerovanie podrobnej správy o audite SOX-COBIT.

Výsledkom bude súhrnná správa, bez ohľadu na to, či použijete voľbu **–p** alebo nie. Súhrnná správa obsahuje informácie o tom, koľko pravidiel bolo spracovaných, koľko bolo neúspešných (našli sa prípady nesúladu so štandardmi) a pre akú úroveň zabezpečenia bol systém kontrolovaný (v tomto prípade to bude SCBPS).

AIX Security Expert - Skupina Password Policy Rules

AIX Security Expert poskytuje špecifické pravidlá pre politiku hesiel.

Silná politika hesiel je jedným zo základných pilierov na dosiahnutie bezpečnosti systému. Politika hesiel zabezpečuje, že heslá sa nedajú ľahko uhádnuť (heslá majú vhodnú kombináciu alfanumerických znakov, číslic a špeciálnych znakov), pravidelne sa im končí platnosť a po skončení platnosti sa už nedajú znova použiť. Nasledujúca tabuľka vypisuje pravidlá politiky hesiel pre každé nastavenie bezpečnosti.

Tabuľka 20. Pravidlá politiky hesiel AIX Security Expert

Názov tlačidla akcie	Definícia	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Minimum number of characters	Nastavuje príslušnú hodnotu atribútu **mindiff** /etc/security/user, ktorá zadáva minimálny počet znakov požadovaných v novom hesle, ktoré neboli v starom hesle.	**Vysoká úroveň bezpečnosti** 4 **Stredná úroveň bezpečnosti** 3 **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Minimum age for password	Nastavuje príslušnú hodnotu atribútu **minage** /etc/security/user, ktorá zadáva minimálny počet týždňov, po uplynutí ktorých sa môže heslo zmeniť.	**Vysoká úroveň bezpečnosti** 1 **Stredná úroveň bezpečnosti** 4 **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Maximum age for password	Nastavuje príslušnú hodnotu atribútu **maxage** /etc/security/user, ktorá zadáva maximálny počet týždňov, po uplynutí ktorých sa môže heslo zmeniť.	**Vysoká úroveň bezpečnosti** 13 **Stredná úroveň bezpečnosti** 13 **Nízka úroveň bezpečnosti** 52 **Štandardné nastavenia AIX** Bez ohraničenia	Áno

Tabuľka 20. Pravidlá politiky hesiel AIX Security Expert (pokračovanie)

Názov tlačidla akcie	Definícia	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Minimum length for password	Nastavuje príslušnú hodnotu atribútu **minlen** /etc/security/user, ktorá zadáva minimálnu dĺžku hesla.	**Vysoká úroveň bezpečnosti** 8 **Stredná úroveň bezpečnosti** 8 **Nízka úroveň bezpečnosti** 8 **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Minimum number of alphabetic characters	Nastavuje príslušnú hodnotu atribútu **minalpha** /etc/security/user, ktorá zadáva minimálny počet abecedných znakov v hesle.	**Vysoká úroveň bezpečnosti** 2 **Stredná úroveň bezpečnosti** 2 **Nízka úroveň bezpečnosti** 2 **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Password reset time	Nastavuje príslušnú hodnotu atribútu **histexpire** /etc/security/user, ktorá zadáva počet týždňov, po uplynutí ktorých je možné heslo resetovať.	**Vysoká úroveň bezpečnosti** 13 **Stredná úroveň bezpečnosti** 13 **Nízka úroveň bezpečnosti** 26 **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Maximum times a char can appear in a password	Nastavuje príslušnú hodnotu atribútu **maxrepeats** /etc/security/user, ktorá zadáva, maximálne koľkokrát sa znak môže v hesle objaviť.	**Vysoká úroveň bezpečnosti** 2 **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** 8	Áno
Password reuse time	Nastavuje príslušnú hodnotu atribútu **histsize** /etc/security/user, ktorá zadáva počet predchádzajúcich hesiel, ktoré užívateľ nemôže znova použiť.	**Vysoká úroveň bezpečnosti** 20 **Stredná úroveň bezpečnosti** 4 **Nízka úroveň bezpečnosti** 4 **Štandardné nastavenia AIX** Bez ohraničenia	Áno

Tabuľka 20. Pravidlá politiky hesiel AIX Security Expert (pokračovanie)

Názov tlačidla akcie	Definícia	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Time to change password after the expiration	Nastavuje príslušnú hodnotu atribútu **maxexpired** /etc/security/user, ktorá zadáva maximálny počet týždňov po **maxage**, kedy môže užívateľ zmeniť heslo s uplynutou platnosťou.	**Vysoká úroveň bezpečnosti** 2 **Stredná úroveň bezpečnosti** 4 **Nízka úroveň bezpečnosti** 8 **Štandardné nastavenia AIX** -1	Áno
Minimum number of non-alphabetic characters	Nastavuje príslušnú hodnotu atribútu **minother** /etc/security/user, ktorá zadáva minimálny počet neabecedných znakov v hesle.	**Vysoká úroveň bezpečnosti** 2 **Stredná úroveň bezpečnosti** 2 **Nízka úroveň bezpečnosti** 2 **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Password expiration warning time	Nastavuje príslušnú hodnotu atribútu **pwdwarntime** /etc/security/user, ktorá zadáva počet dní pred zaslaním varovania systémom o potrebe zmeniť heslo.	**Vysoká úroveň bezpečnosti** 5 **Stredná úroveň bezpečnosti** 14 **Nízka úroveň bezpečnosti** 5 **Štandardné nastavenia AIX** Bez ohraničenia	Áno

AIX Security Expert - Skupina User Group System and Password definitions

AIX Security Expert vykonáva špecifické akcie pre definície užívateľov, skupín a hesiel.

Tabuľka 21. Systém skupiny užívateľov a definície hesiel AIX Security Expert

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Kontrola definícií skupín	Overuje správnosť definícií skupín. Spúšťa nasledujúci príkaz na opravu a hlásenie chýb: `% grpck -y ALL`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Áno **Štandardné nastavenia AIX** Bez účinku	Nie

Tabuľka 21. Systém skupiny užívateľov a definície hesiel AIX Security Expert (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Aktualizácia TCB	Používa príkaz **tcbck** na overenie a aktualizáciu TCB. Spúšťa nasledujúci príkaz: `% tcbck -y ALL` **Poznámka:** Ak je vo vašom systéme vyžadované TCB, toto pravidlo zlyhá, pokiaľ TCB nie je zapnuté. Aj pravidlo nevyhnutnej podmienky zlyhá (prereqtcb) s varovaním. **Nevyhnutná podmienka**: TCB musí byť vybraté pri inštalácii systému.	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Áno **Štandardné nastavenia AIX** Áno	Nie

Kontrola definícií súborov	Používa príkaz **sysck** na kontrolu a opravu bázy súboru /etc/objrepos/inventory:
	`% sysck -i -f \`
	`/etc/security/sysck.cfg.rte`
		Vysoká úroveň bezpečnosti
		Áno
		Stredná úroveň bezpečnosti
		Áno
		Nízka úroveň bezpečnosti
		Áno
		Štandardné nastavenia AIX
		Bez účinku
			Nie

Kontrola definícií hesiel	Overuje správnosť definícií hesiel. Spúšťa nasledujúci príkaz na opravu a hlásenie chýb: `% pwdck -y ALL`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Áno **Štandardné nastavenia AIX** Bez účinku	Nie
Kontrola definícií užívateľov	Overuje správnosť definícií užívateľov. Spúšťa nasledujúci príkaz na opravu a hlásenie chýb: `% usrck -y ALL`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Áno **Štandardné nastavenia AIX** Bez účinku	Nie

AIX Security Expert - Skupina Login Policy Recommendations

AIX Security Expert poskytuje špecifické nastavenia pre politiku prihlasovania.

Poznámka: Na zabezpečenie lepšej sledovateľnosti aktivít súvisiacich s bezpečnosťou vykonávaných užívateľom s oprávneniami typu root sa odporúča, aby sa užívatelia namiesto prihlásenia ako root najprv prihlásili pomocou svojich normálnych ID užívateľa a potom spustili **su** *command* na spustenie príkazov ako užívateľ s oprávneniami typu root. Systém potom môže priradiť rôznych užívateľov k aktivitám vykonávaným pomocou konta root, keď viacerí užívatelia poznajú a používajú heslo root.

Tabuľka 22. Odporúčania pre politiku prihlasovania AIX Security Expert

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Interval between unsuccessful logins	Nastavuje príslušnú hodnotu atribútu **logininterval** /etc/security/login.cfg zadávajúcu časový interval (v sekundách), počas ktorého sa musia uskutočniť neúspešné pokusy o prihlásenie pred vypnutím portu. Ak je napríklad atribút **logininterval** nastavený na 60 a **logindisable** na hodnotu 4, konto bude vypnuté po štyroch neúspešných pokusoch o prihlásenie vykonaných do jednej minúty.	**Vysoká úroveň bezpečnosti** 300 **Stredná úroveň bezpečnosti** 60 **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Number of login attempts before locking the account	Nastavuje príslušnú hodnotu atribútu **loginretries** /etc/security/user, ktorá zadáva počet po sebe idúcich pokusov o prihlásenie na jedno konto pred vypnutím konta. Nenastavujte na root.	**Vysoká úroveň bezpečnosti** 3 **Stredná úroveň bezpečnosti** 4 **Nízka úroveň bezpečnosti** 5 **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Remote root login	Mení hodnotu atribútu **rlogin** /etc/security/user, ktorá zadáva, či je v systéme pre konto root vzdialené prihlásenie zapnuté alebo nie.	**Vysoká úroveň bezpečnosti** False **Stredná úroveň bezpečnosti** False **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** True	Áno
Re-enable login after locking	Nastavuje príslušnú hodnotu atribútu **loginreenable** /etc/security/login.cfg, ktorá zadáva časový interval (v sekundách), po uplynutí ktorého bude port odomknutý po jeho vypnutí atribútom **logindisable**.	**Vysoká úroveň bezpečnosti** 360 **Stredná úroveň bezpečnosti** 30 **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Disable login after unsuccessful login attempts	Nastavuje príslušnú hodnotu atribútu **logindisable** /etc/security/login.cfg, ktorá zadáva počet neúspešných pokusov o prihlásenie na port pred zamknutím portu.	**Vysoká úroveň bezpečnosti** 10 **Stredná úroveň bezpečnosti** 10 **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez ohraničenia	Áno

Tabuľka 22. Odporúčania pre politiku prihlasovania AIX Security Expert (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Login timeout	Nastavuje príslušnú hodnotu atribútu **logintimeout** /etc/security/login.cfg, ktorá zadáva časový interval povolený na napísanie hesla.	**Vysoká úroveň bezpečnosti** 30 **Stredná úroveň bezpečnosti** 60 **Nízka úroveň bezpečnosti** 60 **Štandardné nastavenia AIX** 60	Áno
Delay between unsuccessful logins	Nastavuje príslušnú hodnotu atribútu **logindelay** /etc/security/login.cfg, ktorá zadáva oneskorenie (v sekundách) medzi neúspešnými prihláseniami. Po každom neúspešnom prihlásení bude pridaná ďalšia perióda oneskorenia. Ak je napríklad atribút **logindelay** nastavený na hodnotu 5, terminál bude po prvom neúspešnom prihlásení po ďalšiu požiadavku čakať 5 sekúnd. Po druhom neúspešnom prihlásení bude terminál čakať 10 sekúnd (2*5) a po treťom neúspešnom prihlásení bude terminál čakať 15 sekúnd (3*5).	**Vysoká úroveň bezpečnosti** 10 **Stredná úroveň bezpečnosti** 4 **Nízka úroveň bezpečnosti** 5 **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Local login	Mení hodnotu atribútu **login** /etc/security/user, ktorá zadáva, či je prihlásenie konzoly v systéme pre konto root zapnuté alebo nie.	**Vysoká úroveň bezpečnosti** False **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** True	Áno

AIX Security Expert - Skupina Audit Policy Recommendations

AIX Security Expert poskytuje špecifické nastavenia politiky auditu.

Podobne ako iné nastavenia bezpečnosti aj binárne auditovanie vyžaduje splnenie pravidiel analýzy (nevyhnutných), aby bolo možné použiť pravidlá auditovania pre vysokú, strednú alebo nízku úroveň bezpečnosti. Pre binárne auditovanie musia byť splnené tieto pravidlá analýzy:

1. Nevyhnutné pravidlo pre audit musí zistiť, či audit nie je momentálne spustený. Ak je audit už spustený, potom bol v minulosti nakonfigurovaný a AIX Security Expert nesmie zmeniť existujúcu konfiguráciu a procedúru auditu.
2. Vyžaduje sa minimálne 100 megabajtov voľného priestoru v skupine nosičov, ktorý sa automaticky zapne alebo musí aktuálne existovať súborový systém /audit s veľkosťou minimálne 100 megabajtov.

Ak je vyhovené týmto požiadavkám a voľby auditovania boli nastavené v AIX Security Expert, AIX Security Expert nakonfiguruje a aktivuje auditovanie na systéme nasledujúcim spôsobom. Tlačidlo akcie AIX Security Expert **Enable binaudit** nastavuje politiku auditu. Auditovanie musí byť v systéme zapnuté.

1. Súborový systém JFS /audit musí byť vytvorený a pripojený pred začatím auditu. Súborový systém musí mať veľkosť minimálne 100 megabajtov.
2. Audit musí byť spustený v binárnom režime. Súbor /etc/security/audit/config musí byť nakonfigurovaný takto:

```
start:
            binmode = on
            streammode = off
bin:
            trail = /audit/trail
            bin1 = /audit/bin1
```

```
        bin2 = /audit/bin2
        binsize = 10240
        cmds = /etc/security/audit/bincmds
.
.
atď
```

3. Pridajte položky auditovania pre užívateľa s oprávneniami typu root a bežného užívateľa pre vysokú, strednú a nízku úroveň bezpečnosti.
4. Audit musí byť zapnutý pri opätovnom zavedení pre vysokú, strednú a nízku úroveň bezpečnosti.
5. Noví vytvorení užívatelia musia mať audit zapnutý pre vysokú, strednú a nízku úroveň bezpečnosti, čo možno vykonať pridaním položky auditclasses do stanzy užívateľa v súbore /usr/lib/security/mkuser.default.
6. Aby ste zabránili zaplneniu súborového systému /audit, musíte pridať **cronjob**.

Pravidlo vrátenia auditu musí vypnúť audit a odstrániť jeho zapnutie pri opätovnom zavedení.

Nasledujúce tabuľky uvádzajú hodnoty nastavené pomocou AIX Security Expert pre **Enable binaudit**:

Tabuľka 23. Hodnoty nastavené pomocou AIX Security Expert pre Enable binaudit

Vysoká úroveň bezpečnosti	Stredná úroveň bezpečnosti	Nízka úroveň bezpečnosti	Štandardné nastavenia AIX

Pridajte tieto položky auditovania pre užívateľa s oprávneniami typu root a bežného užívateľa:
`Root:`
`General`
`Src`
`Mail`
`Cron`
`Tcpip`
`Ipsec`
`Lvm`
`User:`
`General`
`Src`
`Cron`
`Tcpip`
 	
Ak chcete povoliť auditovanie novo vytvorených užívateľov, pridajte túto položku do stanzy užívateľa súboru /usr/lib/security/mkuser.default.
`auditclasses=general,SRC,\`
`cron,tcpip`
	Pridajte tieto položky auditovania pre užívateľa s oprávneniami typu root a bežného užívateľa:
	`Root:`
	`General`
	`Src`
	`Tcpip`
	`User:`
	`General`
	`Tcpip`
	
	Ak chcete povoliť auditovanie novo vytvorených užívateľov, pridajte túto položku do stanzy užívateľa súboru /usr/lib/security/mkuser.default.
	`auditclasses=general,`
	`tcpip`
		Pridajte tieto položky auditovania pre užívateľa s oprávneniami typu root a bežného užívateľa:
		`Root:`
		`General`
		`Tcpip`
		`User:`
		`General`
		
		Ak chcete povoliť auditovanie novo vytvorených užívateľov, pridajte túto položku do stanzy užívateľa súboru /usr/lib/security/mkuser.default.
		`auditclasses=general`
			Súbor /etc/security/audit/config obsahuje túto položku:
			`default=login`
			
			Prihlásenie triedy auditu je definované takto:
			`login = USER_SU,`
			`USER_Login,`
			`USER_Logout,`
			`TERM_Logout,`
			`USER_Exit`
			
			Poznámka: Funkcia štandardných nastavení vypne auditovanie.

Tabuľka 23. Hodnoty nastavené pomocou AIX Security Expert pre Enable binaudit (pokračovanie)

Vysoká úroveň bezpečnosti	Stredná úroveň bezpečnosti	Nízka úroveň bezpečnosti	Štandardné nastavenia AIX
Pridajte tieto položky auditovania pre užívateľa s oprávneniami typu root a bežného užívateľa: **root:** general src mail cron tcpip ipsec lvm aixpert *Užívateľ:* general src cron tcpip Ak chcete povoliť auditovanie novo vytvorených užívateľov, pridajte túto položku do stanzy užívateľa súboru /usr/lib/security/mkuser.default. `auditclasses=general,SRC, cron,tcpip`	Pridajte tieto položky auditovania pre užívateľa s oprávneniami typu root a bežného užívateľa: **root:** general src tcpip aixpert *Užívateľ:* general tcpip Ak chcete povoliť auditovanie novo vytvorených užívateľov, pridajte túto položku do stanzy užívateľa súboru /usr/lib/security/mkuser.default. `auditclasses=general, tcpip`	Pridajte tieto položky auditovania pre užívateľa s oprávneniami typu root a bežného užívateľa: **root:** general tcpip aixpert *Užívateľ:* general Ak chcete povoliť auditovanie novo vytvorených užívateľov, pridajte túto položku do stanzy užívateľa súboru /usr/lib/security/mkuser.default. auditclasses=general	Áno

cronjob musí byť spustený každú hodinu a kontrolovať veľkosť /audit. Ak platí Audit Freespace Equation, musia byť vykonané akcie Audit Trail Copy Actions. Audit Freespace Equation definujte s cieľom zabrániť naplneniu súborového systému /audit; ak je súborový systém /audit plný, vykonajú sa akcie Audit Trail Copy Actions (vypnutie auditovania, zálohovanie /audit/trail do /audit/trailOneLevelBack a zapnutie auditovania).

AIX Security Expert - Skupina /etc/inittab Entries

AIX Security Expert označí špecifické položky v súbore /etc/inittab znakom komentára, takže tieto sa pri zavedení systému nespustia.

Tabuľka 24. AIX Security Expert Položky /etc/inittab

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Disable **qdaemon**/Enable **qdaemon**	Komentuje alebo nekomentuje nasledujúcu položku v /etc/inittab.conf: `qdaemon:2:wait:/usr/bin/startsrc –sqdaemon`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Komentár **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Zrušený komentár	Áno

Tabuľka 24. AIX Security Expert Položky /etc/inittab (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Disable **lpd** daemon/Enable **lpd** daemon	Komentuje alebo nekomentuje nasledujúcu položku v /etc/inittab.conf: `lpd:2:once:/usr/bin/startsrc -s lpd`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Komentár **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Zrušený komentár	Áno
Disable CDE/Enable CDE	Ak systém nemá nakonfigurované LFT, komentuje alebo nekomentuje nasledujúcu položku v /etc/inittab: `dt:2:wait:/etc/rc.dt`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Komentár **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Zrušený komentár	Áno
Disable **piobe** daemon/Enable **piobe** daemon	Komentuje alebo nekomentuje nasledujúcu položku v /etc/inittab.conf: `piobe:2:wait:/usr/lib/lpd/pio/etc/pioinit >/dev/null 2>&1`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Komentár **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Zrušený komentár	Áno

AIX Security Expert /etc/rc.tcpip Settings group

AIX Security Expert komentuje špecifické položky v /etc/rc.tcpip, takže tieto sa pri zavedení systému nespustia.

Nasledujúca tabuľka vypisuje položky, ktoré sú komentované v /etc/rc.tcpip, takže tieto sa pri zavedení systému nespustia.

Tabuľka 25. AIX Security Expert /etc/rc.tcpip Nastavenia

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Disable mail client/Enable mail client	Komentuje alebo nekomentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/lib/sendmail "$src_running"`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Zrušený komentár	Áno
Disable routing daemon	Komentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/routed "$src_running" -q`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable **mrouted** daemon	Komentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/mrouted "$src_running"`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable **timed** daemon	Komentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/timed`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Áno **Štandardné nastavenia AIX** Áno	Áno

Tabuľka 25. AIX Security Expert /etc/rc.tcpip Nastavenia (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Disable **rwhod** daemon	Komentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/rwhod "$src_running"`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable print daemon	Komentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/lpd "$src_running"`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable SNMP daemon/Enable SNMP daemon	Komentuje alebo nekomentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/snmpd "$src_running"`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Komentár **Nízka úroveň bezpečnosti** Vypne démona SNMP **Štandardné nastavenia AIX** Zrušený komentár	Áno
Stop DHCP Agent	Komentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/dhcprd "$src_running"`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno

Tabuľka 25. AIX Security Expert /etc/rc.tcpip Nastavenia (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Stop DHCP Server	Komentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/dhcpsd "$src_running"`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Stop autoconf6	Komentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/autoconf6 ""`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable DNS daemon	Komentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/named "$src_running"`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable **gated** daemon	Komentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/gated "$src_running"`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Áno **Štandardné nastavenia AIX** Áno	Áno

Tabuľka 25. AIX Security Expert /etc/rc.tcpip Nastavenia (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Stop DHCP Client	Komentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/dhcpd "$src_running"`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable DPID2 daemon	Komentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/dpid2 "$src_running"`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable NTP daemon	Komentuje nasledujúcu položku z /etc/rc.tcpip: `start /usr/sbin/xntpd "$src_running"`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno

AIX Security Expert - Skupina /etc/inetd.conf Settings

AIX Security Expert označí položky v súbore /etc/inetd.conf znakom komentára.

Predvolená inštalácia AIX povoľuje množstvo sieťových služieb, ktoré môžu ohroziť bezpečnosť systému. AIX Security Expert vypína nepotrebné a nebezpečné služby komentovaním ich príslušných položiek zo súboru /etc/inetd.conf. Pre štandardné nastavenia AIX nie sú tieto položky komentované. Nasledujúca tabuľka vypisuje položky, ktoré sú komentované alebo nekomentované v /etc/inetd.conf.

Tabuľka 26. AIX Security Expert Nastavenia /etc/inetd.conf

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Disable **sprayd** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `sprayd sunrpc_udp udp wait root \ /usr/lib/netsvc/`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable UDP chargen service in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `chargen dgram udp wait root internal`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable telnet / Enable telnet	Komentuje alebo nekomentuje nasledujúcu položku z /etc/inetd.conf: `telnet stream tcp6 nowait root \ /usr/sbin/telnetd telnetd`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Zrušený komentár	Áno
Disable UDP Echo service in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `echo dgram udp wait root internal`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable **tftp** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `tftp dgram udp6 SRC nobody \ /usr/sbin/tftpd tftpd -n`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno

Tabuľka 26. AIX Security Expert Nastavenia /etc/inetd.conf (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Disable **krshd** daemon	Komentuje nasledujúcu položku z /etc/inetd.conf: `kshell stream tcp nowait root \ /usr/sbin/krshd krshd`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable **rusersd** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `rusersd sunrpc_udp udp wait root \ /usr/lib/netsvc/`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable **rexecd** in /etc/inetd.conf / Enable **rexecd** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `exec stream tcp6 nowait root \ /usr/sbin/rexecd rexecd`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Komentár **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Zrušený komentár	Áno
Disable POP3D	Komentuje nasledujúcu položku z /etc/inetd.conf: `pop3 stream tcp nowait root \ /usr/sbin/pop3d pop3d`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable **pcnfsd** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `pcnfsd sunrpc_udp udp wait root \ /usr/sbin/rpc.pcnfsd pcnfsd`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno

Tabuľka 26. AIX Security Expert Nastavenia /etc/inetd.conf (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Disable **bootpd** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `bootps dgram udp wait root \ /usr/sbin/bootpd`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable **rwalld** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `rwalld sunrpc_udp udp wait root \ /usr/lib/netsvc/`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable UDP discard service in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `discard dgram udp wait root \ internal`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable TCP daytime service in /etc/inetd.conf / Enable TCP daytime service in /etc/inetd.conf	Komentuje alebo nekomentuje nasledujúcu položku z /etc/inetd.conf: `daytime stream tcp nowait root \ internal`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Zrušený komentár	Áno
Disable **netstat** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `netstat stream tcp nowait nobody \ /usr/bin/netstat`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno

Tabuľka 26. AIX Security Expert Nastavenia /etc/inetd.conf (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie

Disable **rshd** daemon/Enable **rshd** daemon	Komentuje alebo nekomentuje nasledujúcu položku z /etc/inetd.conf:
	`shell stream tcp6 nowait root \`
	`/usr/sbin/rshd rshd rshd`
		Vysoká úroveň bezpečnosti
		Komentár
		Stredná úroveň bezpečnosti
		Komentár
		Nízka úroveň bezpečnosti
		Komentár
		Štandardné nastavenia AIX
		Zrušený komentár
			Áno

Disable **cmsd** service in /etc/inetd.conf / Enable **cmsd** service in /etc/inetd.conf	Komentuje alebo nekomentuje nasledujúcu položku z /etc/inetd.conf:
	`cmsd sunrpc_udp udp wait root \`
	`/usr/dt/bin/rpc.cms cmsd`
		Vysoká úroveň bezpečnosti
		Komentár
		Stredná úroveň bezpečnosti
		Bez účinku
		Nízka úroveň bezpečnosti
		Bez účinku
		Štandardné nastavenia AIX
		Zrušený komentár
			Áno

Disable **ttdbserver** service in /etc/inetd.conf / Enable **ttdbserver** service in /etc/inetd.conf	Komentuje alebo nekomentuje nasledujúcu položku z /etc/inetd.conf:
	`ttdbserver sunrpc_tcp tcp wait \`
	`root /usr/dt/bin/`
		Vysoká úroveň bezpečnosti
		Komentár
		Stredná úroveň bezpečnosti
		Bez účinku
		Nízka úroveň bezpečnosti
		Bez účinku
		Štandardné nastavenia AIX
		Zrušený komentár
			Áno

Disable **uucpd** in /etc/inetd.conf / Enable **uucpd** in /etc/inetd.conf	Komentuje alebo nekomentuje nasledujúcu položku z /etc/inetd.conf:
	`uucp stream tcp nowait root \`
	`/usr/sbin/uucpd uucpd`
		Vysoká úroveň bezpečnosti
		Komentár
		Stredná úroveň bezpečnosti
		Bez účinku
		Nízka úroveň bezpečnosti
		Bez účinku
		Štandardné nastavenia AIX
		Zrušený komentár
			Áno

Disable UDP time service in /etc/inetd.conf / Enable UDP time service in /etc/inetd.conf	Komentuje alebo nekomentuje nasledujúcu položku z /etc/inetd.conf: `time dgram udp wait root internal`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Zrušený komentár	Áno

Tabuľka 26. AIX Security Expert Nastavenia /etc/inetd.conf (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Disable TCP time service in /etc/inetd.conf / Enable TCP time service in /etc/inetd.conf	Komentuje alebo nekomentuje nasledujúcu položku z /etc/inetd.conf: `time stream tcp nowait root \ internal`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Zrušený komentár	Áno
Disable **rexd** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `rexd sunrpc_tcp tcp wait root \ /usr/sbin/tpc.rexd.rexd rexd`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Áno **Štandardné nastavenia AIX** Áno	Áno
Disable TCP chargen service in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `chargen stream tcp nowait root \ internal`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable **rlogin** in /etc/inetd.conf / Enable **rlogin** in /etc/inetd.conf	Komentuje alebo nekomentuje nasledujúcu položku z /etc/inetd.conf: `login stream tcp6 nowait root \ /usr/sbin/rlogind rlogind`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Komentár **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Zrušený komentár	Áno
Disable talk in /etc/inetd.conf	Komentuje alebo nekomentuje nasledujúcu položku z /etc/inetd.conf: `talk dgram udp wait root \ /usr/sbin/talkd talkd`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Komentár **Nízka úroveň bezpečnosti** Komentár **Štandardné nastavenia AIX** Zrušený komentár	Áno

Tabuľka 26. AIX Security Expert Nastavenia /etc/inetd.conf (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Disable **fingerd** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `finger stream tcp nowait nobody \ /usr/sbin/fingerd fingerd`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable FTP / Enable FTP	Komentuje alebo nekomentuje nasledujúcu položku z /etc/inetd.conf: `ftp stream tcp6 nowait root \ /usr/sbin/ftpd ftpd`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Zrušený komentár	Áno
Disable IMAPD	Komentuje nasledujúcu položku z /etc/inetd.conf: `imap2 stream tcp nowait root \ /usr/sbin/imapd imapd`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable **comsat** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `comsat dgram udp wait root \ /usr/sbin/comsat comsat`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable **rquotad** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `rquotad sunrpc_udp udp wait root \ /usr/sbin/rpc.rquotad`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Áno **Štandardné nastavenia AIX** Áno	Áno

Tabuľka 26. AIX Security Expert Nastavenia /etc/inetd.conf (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Disable UDP daytime service in /etc/inetd.conf / Enable UDP daytime service in /etc/inetd.conf	Komentuje alebo nekomentuje nasledujúcu položku z /etc/inetd.conf: `daytime dgram udp wait root internal`	**Vysoká úroveň bezpečnosti** Komentár **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Zrušený komentár	Áno

Disable **krlogind** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf:
	`klogin stream tcp nowait root \`
	`/usr/sbin/krlogind krlogind`
		Vysoká úroveň bezpečnosti
		Áno
		Stredná úroveň bezpečnosti
		Bez účinku
		Nízka úroveň bezpečnosti
		Bez účinku
		Štandardné nastavenia AIX
		Áno
			Áno

Disable TCP Discard service in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf:
	`discard stream tcp nowait root \`
	`internal`
		Vysoká úroveň bezpečnosti
		Áno
		Stredná úroveň bezpečnosti
		Bez účinku
		Nízka úroveň bezpečnosti
		Bez účinku
		Štandardné nastavenia AIX
		Áno
			Áno

Disable TCP echo service in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `echo stream tcp nowait root internal`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno

Disable **sysstat** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf:
	`systat stream tcp nowait nodby \`
	`/usr/bin/ps ps -ef`
		Vysoká úroveň bezpečnosti
		Áno
		Stredná úroveň bezpečnosti
		Áno
		Nízka úroveň bezpečnosti
		Bez účinku
		Štandardné nastavenia AIX
		Áno
			Áno

Tabuľka 26. AIX Security Expert Nastavenia /etc/inetd.conf (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Disable **rstatd** in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `rstatd sunrpc_udp udp wait root \ /usr/sbin/rpc.rstatd rstatd`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable dtspc in /etc/inetd.conf	Komentuje nasledujúcu položku z /etc/inetd.conf: `dtspc stream tcp nowait root \ /usr/dt/bin/dtspcd`	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno

AIX Security Expert - Skupina Disable SUID of Commands

Nasledujúce príkazy sú štandardne nainštalované s bitovou sadou SUID. Tento bit nie je nastavený pre vysokú, strednú a nízku bezpečnosť. Pre štandardné nastavenia AIX je bit SUID na týchto príkazoch obnovený.

Tabuľka 27. AIX Security Expert Disable SUID of Commands

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
hls_filepermgr	Manažér oprávnení súborov: Spúšťa príkaz **fpm** s voľbou high na odstránenie setuid a setgid z privilegovaných príkazov	Vysoká úroveň bezpečnosti	Áno
mls_filepermgr	Manažér oprávnení súborov: Spúšťa príkaz **fpm** s voľbou medium na odstránenie setuid a setgid z privilegovaných príkazov	Stredná úroveň bezpečnosti	Áno
lls_filepermgr	Manažér oprávnení súborov: Spúšťa príkaz **fpm** s voľbou low na odstránenie setuid a setgid z privilegovaných príkazov	Nízka úroveň bezpečnosti	Áno

AIX Security Expert - Skupina Disable Remote Services

AIX Security Expert zakazuje nezabezpečené príkazy pre vysokú a strednú úroveň bezpečnosti.

Nasledujúce príkazy a démony sú často využívané na vyhľadanie medzier v bezpečnosti. Tieto nezabezpečené príkazy odmietajú spúšťať oprávnenia pre vysokú a strednú úroveň bezpečnosti a démony sú vypnuté. Pre nízku úroveň bezpečnosti nie sú tieto príkazy a démony ovplyvnené. Pre štandardné nastavenia AIX sú tieto príkazy a démony zapnuté na použitie.

- **rcp**
- **rlogin**
- **rsh**
- **tftp**
- **rlogind**
- **rshd**

- **tftpd**

Tabuľka 28. AIX Security Expert Disable Remote Services

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Enable unsecure daemons	Ak je TCB zapnuté, nastavuje oprávnenia na spúšťanie démonov **rlogind**, **rshd** a **tftpd** a aktualizuje databázu sysck zmenami bitov režimu pre tieto démony. Ak TCB nie je zapnuté, nastavené sú oprávnenia na spúšťanie démonov **rlogind**, **rshd** a **tftpd**.	**Vysoká úroveň bezpečnosti** Bez účinku **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez účinku	Áno
Zakázať nezabezpečené príkazy	1. Ak je TCB zapnuté, odstraňuje oprávnenia na spúšťanie príkazov **rcp**, **rlogin**, **rsh** a **tftp** a aktualizuje databázu **sysck** zmenami bitov režimu týchto príkazov. Ak TCB nie je zapnuté, odstraňuje oprávnenia na spúšťanie na príkazoch **rcp**, **rlogin** a **rsh**. 2. Zastavuje aktuálne inštancie príkazov **rcp**, **rlogin**, **rsh**, **tftp** a **uftp**, pokým jeden z týchto príkazov nie je rodičovským procesom AIX Security Expert. 3. Pridáva stanzu **tcpip:** do /etc/security/config s cieľom obmedziť použitie **.netrc** v **ftp** a **rexec**.	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez účinku	Áno
Enable unsecure commands	1. Ak je TCB zapnuté, nastavuje oprávnenia na spúšťanie príkazov **rcp**, **rlogin**, **rsh** a **tftp** a aktualizuje databázu **sysck** zmenami bitov režimu týchto príkazov. Ak TCB nie je zapnuté, nastavuje oprávnenia na spúšťanie na príkazoch **rcp**, **rlogin** a **rsh**. 2. Odstraňuje súbor /etc/security/config.	**Vysoká úroveň bezpečnosti** Bez účinku **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Disable unsecure daemons	1. Ak je TCB zapnuté, odstraňuje oprávnenia na spúšťanie démonov **rlogind**, **rshd** a **tftpd** a aktualizuje databázu **sysck** zmenami bitov režimu týchto démonov. Ak TCB nie je zapnuté, odstraňuje oprávnenia na spúšťanie démonov **rlogind**, **rshd** a **tftpd**. 2. Zastavuje aktuálne inštancie démonov **rlogind**, **rshd** a **tftpd**, pokiaľ jeden z týchto démonov nie je rodičovským procesom AIX Security Expert.	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez účinku	Áno
Stop NFS daemon	• Odstraňuje všetky pripojenia NFS • Zakazuje NFS • Odstraňuje skript spustenia NFS z /etc/inittab	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez účinku	Áno

Tabuľka 28. AIX Security Expert Disable Remote Services (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Enable NFS daemon	• Exportuje všetky položky uvedené v /etc/exports • Pridáva položku do /etc/inittab na spustenie /etc/rc.nfs pri systémovom reštarte • Okamžite spúšťa /etc/rc.nfs	**Vysoká úroveň bezpečnosti** Bez účinku **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno

AIX Security Expert - Skupina Remove access that does not require Authentication

AIX podporuje niekoľko služieb, ktoré na prihlásenie do siete nevyžadujú autentifikáciu užívateľa.

Súbor /etc/hosts.equiv a všetky lokálne súbory **$HOME**/.rhosts definujú hostiteľov a kontá užívateľov, ktoré môžu spúšťať vzdialené príkazy na lokálnom hostiteľovi bez hesla. Pokiaľ sa táto schopnosť nevyžaduje explicitne, tieto súbory môžete vymazať.

Tabuľka 29. AIX Security Expert Odstrániť prístup, ktorý nevyžaduje autentifikáciu

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Odstrániť služby rhosts a netrc	Súbory .rhosts a .netrc ukladajú mená užívateľov a heslá v zrozumiteľnom textovom formáte.	**Vysoká úroveň bezpečnosti** Odstrániť súbory .rhosts a .netrc z domovských adresárov všetkých užívateľov vrátane užívateľa s oprávneniami typu root. **Stredná úroveň bezpečnosti** Odstrániť súbory .rhosts a .netrc z domovských adresárov všetkých užívateľov vrátane užívateľa s oprávneniami typu root. **Nízka úroveň bezpečnosti** Odstrániť súbory .rhosts a .netrc z domovského adresára užívateľa s oprávneniami typu root. **Štandardné nastavenia AIX** Odstrániť súbory .rhosts a .netrc z domovských adresárov všetkých užívateľov vrátane užívateľa s oprávneniami typu root.	Áno
Odstrániť položky zo súboru /etc/hosts.equiv	Súbor /etc/hosts.equiv spolu so súborom lokálneho užívateľa **$HOME**/.rhosts definuje, ktorí užívatelia na cudzích hostiteľoch majú oprávnenie vzdialene spúšťať príkazy na lokálnom hostiteľovi. Ak sa niekto na cudzom hostiteľovi dozvie podrobnosti o mene užívateľa a hostiteľa, títo môžu nájsť spôsob ako spúšťať vzdialené príkazy na lokálnom hostiteľovi bez autentifikácie.	**Vysoká úroveň bezpečnosti** Odstrániť všetky položky zo súboru /etc/hosts.equiv. **Stredná úroveň bezpečnosti** Odstrániť všetky položky zo súboru /etc/hosts.equiv. **Nízka úroveň bezpečnosti** Odstrániť všetky položky zo súboru /etc/hosts.equiv. **Štandardné nastavenia AIX** Odstrániť všetky položky zo súboru /etc/hosts.equiv.	Áno

AIX Security Expert - Skupina Tuning Network Options

Voľby ladenia siete na vhodné hodnoty tvoria významnú súčasť bezpečnosti. Nastavenie sieťového atribútu na 0 vypína túto voľbu a jeho nastavenie na 1 ju zapína.

Nasledujúca tabuľka vypisuje nastavenia sieťových atribútov pre vysokú, strednú a nízku úroveň bezpečnosti. Táto tabuľka poskytuje aj opis spôsobu, akým navrhovaná hodnota ľubovoľnej voľby určitej siete zaisťuje bezpečnosť tejto siete.

Tabuľka 30. AIX Security Expert Voľby ladenia siete pre sieťovú bezpečnosť

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Sieťová voľba ipsrcrouteforward	Zadáva, či systém preposiela pakety smerované zdrojom alebo nie. Vypnutie ipsrcrouteforward zabráni prístupu prostredníctvom zdroja smerujúceho útoky.	**Vysoká úroveň bezpečnosti** 0 **Stredná úroveň bezpečnosti** 0 **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** 1	Áno
Sieťová voľba ipignoreredirects	Zadáva, či sa majú spracúvať prijaté presmerovania alebo nie.	**Vysoká úroveň bezpečnosti** 1 **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Sieťová voľba clean_partial_conns	Zadáva, či sa vyhnúť útokom znakov synchronizácie (SYN) alebo nie.	**Vysoká úroveň bezpečnosti** 1 **Stredná úroveň bezpečnosti** 1 **Nízka úroveň bezpečnosti** 1 **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Sieťová voľba ipsrcrouterecv	Zadáva, či systém akceptuje pakety smerované zdrojom alebo nie. Zakázanie ipsrcrouterecv zabráni prístupu prostredníctvom zdroja smerujúceho útoky.	**Vysoká úroveň bezpečnosti** 0 **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez ohraničenia	Áno

Tabuľka 30. AIX Security Expert Voľby ladenia siete pre sieťovú bezpečnosť (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Sieťová voľba ipforwarding	Zadáva, či má jadro preposielať pakety alebo nie. Vypnutím voľby ipforwarding sa zabráni tomu, aby presmerované pakety dosiahli vzdialenú sieť.	**Vysoká úroveň bezpečnosti** 0 **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Sieťová voľba ipsendredirects	Zadáva, či má jadro zaslať signály presmerovania alebo nie. Vypnutie voľby ipsendredirects zabráni tomu, aby presmerované pakety dosiahli vzdialenú sieť.	**Vysoká úroveň bezpečnosti** 0 **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** 1	Áno
Sieťová voľba ip6srcrouteforward	Zadáva, či systém preposiela pakety IPv6 smerované zdrojom alebo nie. Vypnutie voľby ip6srcrouteforward zabráni prístupu prostredníctvom zdroja smerujúceho útoky.	**Vysoká úroveň bezpečnosti** 0 **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** 1	Áno
Sieťová voľba directed_broadcast	Zadáva, či povoliť smerované vysielanie do brány alebo nie. Vypnutie voľby directed_broadcast pomáha zabrániť tomu, aby smerované pakety dosiahli vzdialenú sieť.	**Vysoká úroveň bezpečnosti** 0 **Stredná úroveň bezpečnosti** 0 **Nízka úroveň bezpečnosti** 0 **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Sieťová voľba tcp_pmtu_discover	Vypne alebo zapne zisťovanie veľkosti paketov (path MTU discovery) pre aplikácie využívajúce protokol TCP. Vypnutie voľby tcp_pmtu_discover zabráni prístupu prostredníctvom zdroja smerujúceho útoky.	**Vysoká úroveň bezpečnosti** 0 **Stredná úroveň bezpečnosti** 0 **Nízka úroveň bezpečnosti** 0 **Štandardné nastavenia AIX** 1	Áno

Tabuľka 30. AIX Security Expert Voľby ladenia siete pre sieťovú bezpečnosť (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Sieťová voľba bcastping	Povoľuje odpoveď na pakety echa ICMP zaslané na vysielaciu adresu. Vypnutie voľby bcastping zabráni útokom smurf.	**Vysoká úroveň bezpečnosti** 0 **Stredná úroveň bezpečnosti** 0 **Nízka úroveň bezpečnosti** 0 **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Sieťová voľba icmpaddressmask	Zadáva, či systém odpovedá na požiadavku masky adresy ICMP alebo nie. Vypnutie voľby icmpaddressmask zabráni prístupu prostredníctvom zdroja smerujúceho útoky.	**Vysoká úroveň bezpečnosti** 0 **Stredná úroveň bezpečnosti** 0 **Nízka úroveň bezpečnosti** 0 **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Sieťová voľba udp_pmtu_discover	Zapína alebo vypína zistenie MTU (maximum transfer unit) cesty pre aplikácie UDP. Vypnutie voľby udp_pmtu_discover zabráni prístupu prostredníctvom zdroja smerujúceho útoky.	**Vysoká úroveň bezpečnosti** 0 **Stredná úroveň bezpečnosti** 0 **Nízka úroveň bezpečnosti** 0 **Štandardné nastavenia AIX** 1	Áno
Sieťová voľba ipsrcroutesend	Zadáva, či môžu aplikácie preposielať pakety smerované zdrojom alebo nie. Vypnutie voľby ipsrcroutesend zabráni prístupu prostredníctvom zdroja smerujúceho útoky.	**Vysoká úroveň bezpečnosti** 0 **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** 1	Áno
Sieťová voľba nonlocsrcroute	Zadáva do internetového protokolu, či pakety prísne smerované zdrojom môžu byť adresované hostiteľom mimo lokálnej siete alebo nie. Vypnutie voľby nonlocsrcroute zabráni prístupu prostredníctvom zdroja smerujúceho útoky.	**Vysoká úroveň bezpečnosti** 0 **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez ohraničenia	Áno

Tabuľka 30. AIX Security Expert Voľby ladenia siete pre sieťovú bezpečnosť (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Sieťová voľba tcp_tcpsecure	Chráni spojenia TCP pred zraniteľnosťou. Hodnoty: • 0 = žiadna ochrana • 1 = poslanie falošného SYN do vytvoreného pripojenia • 2 = poslanie falošného RST do vytvoreného pripojenia • 3 = vsunutie údajov do vytvoreného TCP pripojenia • 5-7 = kombinácia vyššie uvedených akcií	**Vysoká úroveň bezpečnosti** 7 **Stredná úroveň bezpečnosti** 7 **Nízka úroveň bezpečnosti** 5 **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Sieťová voľba sockthresh	Špecifikuje limit využitia sieťovej pamäte. Žiadne nové soketové pripojenia nemôžu prekročiť hodnotu sockthresh. Udáva maximálnu veľkosť sieťovej pamäte, ktorá môže byť alokovaná pre sokety.	**Vysoká úroveň bezpečnosti** 60 **Stredná úroveň bezpečnosti** 70 **Nízka úroveň bezpečnosti** 85 **Štandardné nastavenia AIX** Bez ohraničenia	Áno

Nasledujúce sieťové voľby súvisia skôr s výkonom siete ako s jej zabezpečením.

Tabuľka 31. AIX Security Expert Ladenie sieťových volieb pre sieťový výkon

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Sieťová voľba rfc1323	Laditeľná voľba rfc1323 zapína voľbu škálovania okna TCP.	**Vysoká úroveň bezpečnosti** 1 **Stredná úroveň bezpečnosti** 1 **Nízka úroveň bezpečnosti** 1 **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Sieťová voľba tcp_sendspace	Laditeľná voľba tcp_sendspace zadáva, koľko údajov môže zasielajúca aplikácia uložiť do vyrovnávacej pamäte v jadre predtým, než bude aplikácia zablokovaná pri volaní odoslania.	**Vysoká úroveň bezpečnosti** 262144 **Stredná úroveň bezpečnosti** 262144 **Nízka úroveň bezpečnosti** 262144 **Štandardné nastavenia AIX** 16384	Áno
Sieťová voľba tcp_mssdflt	Predvolená maximálna veľkosť segmentu používaná v komunikácii so vzdialenými sieťami.	**Vysoká úroveň bezpečnosti** 1448 **Stredná úroveň bezpečnosti** 1448 **Nízka úroveň bezpečnosti** 1448 **Štandardné nastavenia AIX** 1460	Áno

Tabuľka 31. AIX Security Expert Ladenie sieťových volieb pre sieťový výkon (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Sieťová voľba extendednetstats	Povoľuje rozsiahlejšiu štatistiku pre sieťové pamäťové služby.	**Vysoká úroveň bezpečnosti** 1 **Stredná úroveň bezpečnosti** 1 **Nízka úroveň bezpečnosti** 1 **Štandardné nastavenia AIX** Bez ohraničenia	Áno
Sieťová voľba tcp_recvspace	Laditeľná voľba tcp_recvspace zadáva, koľko bajtov údajov môže prijímajúci systém uložiť do vyrovnávacej pamäte v jadre pri prijímaní frontu soketov.	**Vysoká úroveň bezpečnosti** 262144 **Stredná úroveň bezpečnosti** 262144 **Nízka úroveň bezpečnosti** 262144 **Štandardné nastavenia AIX** 16384	Áno
Sieťová voľba sb_max	Laditeľná voľba sb_max nastavuje horný limit na počet vyrovnávacích pamätí soketu vo fronte na jednotlivý soket, ktorý kontroluje, koľko priestoru vyrovnávacej pamäte spotrebujú vyrovnávacie pamäte vo fronte na soket odosielateľa alebo na soket prijímateľa.	**Vysoká úroveň bezpečnosti** 1048576 **Stredná úroveň bezpečnosti** 1048576 **Nízka úroveň bezpečnosti** 1048576 **Štandardné nastavenia AIX** 1048576	Áno

AIX Security Expert - Skupina IPsec filter rules

AIX Security Expert poskytuje nasledujúce filtre IPsec.

Tabuľka 32. Pravidlá filtrovania AIX Security Expert IPsec

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Shun host for 5 minutes	Obíde alebo zablokuje pakety určené pre niekoľko portov tcp a udp so známymi ohrozeniami na hostiteľovi na päť minút. Hostiteľ nebude päť minút prijímať žiadne pakety pre tieto porty.	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez účinku	Áno

Tabuľka 32. Pravidlá filtrovania AIX Security Expert IPsec (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Guard host against port scans	Chráni pred skenovaním portov. Každý vzdialený hostiteľ vykonávajúci skenovanie portu bude na päť minút obídený alebo zablokovaný. Z tohto vzdialeného hostiteľa nebudú päť minút prijímané žiadne pakety.	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez účinku	Áno

AIX Security Expert - Skupina Miscellaneous

AIX Security Expert poskytuje rôzne nastavenia bezpečnosti pre vysokú, strednú a nízku úroveň bezpečnosti.

Tabuľka 33. Skupina Rôzne AIX Security Expert

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Remove dot from path root	Skontroluje, či premenná prostredia PATH v súboroch **$HOME**/.profile, **$HOME**/.kshrc, **$HOME**/.cshrc a **$HOME**/.login neobsahuje bodky (.), a ak áno, odstráni ju. **Poznámka:** Odstránenie bodiek sa vykoná iba vtedy, ak položka v súbore začína premennou prostredia PATH a obsahuje bodky (.). Súbor sa nezmení, ak premenná prostredia PATH obsahuje iné premenné alebo je nastavený na hodnotu vrátenú z programu, ktorý sa volá prostredníctvom skriptu. Príklad cesty, ktorá by sa nezmenila (*pathprog* predstavuje program, ktorý vráti reťazec cesty): `PATH="$(pathprog)"` Z tejto cesty sa bodky odstránia z cesty pred interpretáciou obsahu premennej *pathprog*, takže sa neodstránia žiadne bodky z vrátenej cesty.	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Áno **Štandardné nastavenia AIX** Áno	Áno
Limit system access	Zabezpečuje, že užívateľ s oprávneniami typu root je jediným užívateľom, ktorý má oprávnenie spúšťať úlohy **cron**.	**Vysoká úroveň bezpečnosti** Zabezpečuje, že užívateľ s oprávneniami typu root je jediným užívateľom v súbore cron.allow a odstraňuje súbor cron.deny. **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Odstraňuje súbor cron.allow a vymazáva všetky položky v súbore cron.deny.	Áno

Tabuľka 33. Skupina Rôzne AIX Security Expert (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Remove dot from /etc/environment	Odstráni bodky (.) z premennej prostredia **PATH** v súbore /etc/environment.	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Áno **Štandardné nastavenia AIX** Áno	Áno
Remove dot from non-root path	Odstráni bodky (.) z premennej prostredia **PATH** v súboroch **$HOME**/.profile, **$HOME**/.kshrc, **$HOME**/.cshrc a **$HOME**/.login všetkých užívateľov iných ako root. **Poznámka:** Odstránenie bodiek sa vykoná iba vtedy, ak položka v súbore začína premennou prostredia PATH a obsahuje bodky (.). Súbor sa nezmení, ak premenná prostredia PATH obsahuje iné premenné alebo je nastavený na hodnotu vrátenú z programu, ktorý sa volá prostredníctvom skriptu. Príklad cesty, ktorá by sa nezmenila (*pathprog* predstavuje program, ktorý vráti reťazec cesty): `PATH="$(pathprog)"` Z tejto cesty sa bodky odstránia z cesty pred interpretáciou obsahu premennej *pathprog*, takže sa neodstránia žiadne bodky z vrátenej cesty.	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez účinku	Nie
Add root user in /etc/ftpusers file	Pridáva meno užívateľa s oprávneniami typu root do súboru /etc/ftpusers s cieľom vypnúť **ftp** vzdialeného užívateľa s oprávneniami typu root.	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno
Remove root user in /etc/ftpusers file	Odstraňuje položku užívateľa s oprávneniami typu root z /etc/ftpusers s cieľom zapnúť **ftp** vzdialeného užívateľa s oprávneniami typu root.	**Vysoká úroveň bezpečnosti** Bez účinku **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Áno	Áno

Tabuľka 33. Skupina Rôzne AIX Security Expert (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie

Set login herald	Označuje /etc/security/login.cfg s cieľom zabezpečiť, aby nebola zadaná heraldická hodnota. Ak sa použije predvolená heraldika, heraldika by sa mala zmeniť. Heraldickú hodnotu je možné zmeniť iba vtedy, ak miestne nastavenie systému je **en_US** alebo iné anglické miestne nastavenie. Ak je toto kritérium splnené, hodnota heraldického atribútu v predvolenej stanze súboru /etc/security/login.cfg bude nastavená na:
	`Unauthorized use of this \`
	`system is prohibited.\nlogin:`
	
	Poznámka: Nastavenie bezpečnosti nadobudne účinnosť len pre nové relácie. Nastavenie bezpečnosti nenadobúda účinnosť v relácii, v ktorej bola konfigurácia nastavená.
		Vysoká úroveň bezpečnosti
		herald="Unauthorized use of this system is prohibited.\nlogin:"
		Stredná úroveň bezpečnosti
		herald="Unauthorized use of this system is prohibited.\nlogin:"
		Nízka úroveň bezpečnosti
		herald="Unauthorized use of this system is prohibited.\nlogin:"
		Štandardné nastavenia AIX
		herald=
			Áno

Remove guest account	V prípade úrovne zabezpečenia High, Medium a Low odstráni z počítača konto hosťa, ako aj všetky údaje hosťa. Pre štandardné nastavenie AIX je v systéme vytvorené konto hosťa. **Poznámka:** Administrátor systému musí nastaviť heslo pre toto konto explicitne, pretože AIX Security Expert nie je navrhnutý na spracovanie takýchto užívateľských interaktívnych úloh.	**Vysoká úroveň bezpečnosti** Remove guest account and data **Stredná úroveň bezpečnosti** Remove guest account and data **Nízka úroveň bezpečnosti** Remove guest account and data **Štandardné nastavenia AIX** Pridáva konto hosťa na počítači.	Áno
Crontab permissions	Zabezpečuje, že úlohy užívateľa s oprávneniami typu root **crontab** sú len vo vlastníctve užívateľa s oprávneniami typu root a môže ich zapisovať len užívateľ s oprávneniami typu root.	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Áno **Štandardné nastavenia AIX** Bez účinku	Áno
Enable X-Server access	Prideľuje autentifikáciu pre prístup do X-Server.	**Vysoká úroveň bezpečnosti** Vyžaduje sa autentifikácia **Stredná úroveň bezpečnosti** Vyžaduje sa autentifikácia **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Nie sú potrebné	Nie

Tabuľka 33. Skupina Rôzne AIX Security Expert (pokračovanie)

Názov tlačidla akcie	Opis	Hodnota, ktorú nastavil AIX Security Expert	Odvolanie akcie
Object creation permissions	Nastavuje príslušnú hodnotu atribútu **umask** /etc/security/user, ktorá zadáva predvolené oprávnenia na vytvorenie objektu.	**Vysoká úroveň bezpečnosti** 077 **Stredná úroveň bezpečnosti** 027 **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** 022	Áno
Set core file size	Nastavuje príslušnú hodnotu atribútu **core** /etc/security/limits, ktorá zadáva veľkosť súboru jadra pre root. **Poznámka:** Nastavenie bezpečnosti nadobudne účinnosť len pre nové relácie. Nastavenie bezpečnosti nenadobúda účinnosť v relácii, v ktorej bola konfigurácia nastavená.	**Vysoká úroveň bezpečnosti** 0 **Stredná úroveň bezpečnosti** 0 **Nízka úroveň bezpečnosti** 0 **Štandardné nastavenia AIX** 2097151	Áno
Aktivovať funkciu SED	Aktivuje funkciu Stack Execution Disable a spustí príkaz **sedmgr** na zadaných súboroch. **Poznámka:** Toto pravidlo nadobudne platnosť po reboote systému.	**Vysoká úroveň bezpečnosti** setidfiles **Stredná úroveň bezpečnosti** Bez účinku **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez účinku	
Kontrola integrity hesla užívateľ Root	Zabezpečuje, že heslo užívateľa root nie je slabé. Atribút dictionlist užívateľa root je nastavený na /etc/security/aixpert/dictionary/English, takže príkaz **passwd** dokáže zabezpečiť, že nastavované heslo užívateľa root nebude slabé.	**Vysoká úroveň bezpečnosti** Áno **Stredná úroveň bezpečnosti** Áno **Nízka úroveň bezpečnosti** Bez účinku **Štandardné nastavenia AIX** Bez účinku	Áno

AIX Security - Zrušenie nastavení bezpečnosti

Môžete zrušiť niektoré pravidlá a nastavenia bezpečnosti AIX Security Expert.

Nasledujúce pravidlá a nastavenia bezpečnosti AIX Security Expert nie je možné vziať späť:

- Kontrola definícií hesiel pre vysokú, strednú a nízku úroveň bezpečnosti
- Kontrola definícií užívateľov pre vysokú, strednú a nízku úroveň bezpečnosti
- Kontrola definícií skupín pre vysokú, strednú a nízku úroveň bezpečnosti
- Aktualizácia TCB pre vysokú, strednú a nízku úroveň bezpečnosti
- Zapnutie prístupu na server X pre vysokú, strednú a nízku úroveň bezpečnosti
- Odstránenie bodky z nekoreňovej cesty pre vysokú úroveň bezpečnosti a štandardné nastavenia AIX
- Odstránenie konta hosťa pre vysokú, strednú a nízku úroveň bezpečnosti

Voľba Check Security AIX Security Expert

AIX Security Expert môže vygenerovať správy o aktuálnych nastaveniach systému a sieťovej bezpečnosti.

Po konfigurácii systému pomocou AIX Security Expert (príkaz aixpert) je možné použiť voľbu Check Security na vytvorenie správy o rozličných konfiguračných nastaveniach. Ak bolo ktorékoľvek z týchto nastavení zmenené mimo kontroly AIX Security Expert, voľba AIX Security Expert Check Security zaprotokoluje tieto rozdiely do súboru /etc/security/aixpert/check_report.txt.

Keď napríklad používate nízku úroveň bezpečnosti, démon **talkd** je v /etc/inetd.conf vypnutý. Ak bude démon **talkd** neskôr zapnutý a potom sa spustí voľba Check Security, tieto informácie budú zaprotokolované do súboru check_report.txt nasledovne:

```
coninetdconf.ksh: Servisný rozhovor pomocou protokolového udp by mal byť vypnutý, ale je momentálne zapnutý.
```

Ak neboli použité nastavenia bezpečnosti zmenené, súbor check_report.txt bude prázdny.

Voľbu Check Security spúšťajte pravidelne a ak chcete zistiť, či boli od použitia nastavení bezpečnosti AIX Security Expert nejaké nastavenia zmenené, pozrite si výslednú správu. Voľbu Check Security spúšťajte ako súčasť hlavnej systémovej zmeny, napríklad ako súčasť inštalácie alebo aktualizácie softvéru.

Súvisiace informácie:

príkaz aixpert

Súbory AIX Security Expert

AIX Security Expert vytvára a používa niekoľko súborov.

/etc/security/aixpert/core/aixpertall.xml
: Obsahuje zoznam XML všetkých možných nastavení bezpečnosti.

/etc/security/aixpert/core/appliedaixpert.xml
: Obsahuje zoznam XML použitých nastavení bezpečnosti.

/etc/security/aixpert/core/secaixpert.xml
: Obsahuje zoznam XML vybratých nastavení bezpečnosti spracovaných pomocou AIX Security Expert GUI.

/etc/security/aixpert/log/aixpert.log
: Obsahuje protokol sledovania použitých nastavení bezpečnosti. AIX Security Expert nepoužíva systlog; AIX Security Expert zapisuje priamo do /etc/security/aixpert/log/aixpert.log.

Poznámka: Protokolové súbory a AIX Security Expert XML sú vytvorené s týmito oprávneniami:

/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-------

Scenár vysokej úrovne bezpečnosti AIX Security Expert

Ide o scenár vysokej úrovne bezpečnosti AIX Security Expert.

Zobrazenie AIX Security Expert úrovní bezpečnosti je sčasti odvodené od publikácie National Institute of Standards and Technology *Security Configuration Checklists Progarm for IT Pruducts - Guidance for CheckLists Users and Developers* (názov publikácie hľadajte na webovej stránke NIS: http://www.nist.gov/index.html). Avšak vysoká, stredná a nízka úroveň bezpečnosti znamená pre každého niečo iné. Dôležité je pochopiť prostredie, v ktorom váš systém pracuje. Ak si vyberiete príliš vysokú úroveň bezpečnosti, môže sa stať, že sa do svojho počítača nebudete môcť dostať. Ak si vyberiete príliš nízku úroveň bezpečnosti, váš počítač môže byť ohrozený kybernetickým útokom.

Toto je príklad prostredia vyžadujúceho vysokú úroveň bezpečnosti. Bob bude používať svoj systém spolu s poskytovateľom internetovej služby. Systém bude priamo pripojený na internet, bude spustený ako server HTTP, bude obsahovať citlivé užívateľské údaje a Bob ho bude musieť riadiť vzdialene. Systém by mal byť nastavený a testovaný na izolovanej lokálnej sieti ešte pred jeho on-line pripojením poskytovateľom internetovej služby.

Vysoká úroveň bezpečnosti je pre toto prostredie správna, ale Bob potrebuje vzdialený prístup do systému. Vysoká úroveň bezpečnosti nepovoľuje **telnet**, **rlogin**, **ftp** a iným bežným pripojeniam zasielať heslá cez sieť, pretože by ich niekto na internete mohol ľahko zistiť. Bob potrebuje bezpečnú metódu na vzdialené prihlásenie, napríklad **openssh**. Môže si prečítať úplnú dokumentáciu AIX Security Expert a zistiť, či je v jeho prostredí niečo jedinečné, čomu by mohla vysoká úroveň bezpečnosti zabrániť. Ak áno, môže zrušiť výber, keď sa zobrazí podrobný panel vysokej úrovne bezpečnosti. Bob by mal tiež nakonfigurovať a spustiť server HTTP alebo iné služby, ktoré chce ponúkať vo svojom systéme.

Keď si potom vyberie vysokú úroveň bezpečnosti, AIX Security Expert rozozná, či sú spustené služby potrebné a nebude blokovať prístup k ich portom. Prístup k všetkým ostatným portom môže byť ohrozený a vysoká úroveň bezpečnosti ich zablokuje. Po otestovaní tejto konfigurácie je Bobov počítač pripravený na pripojenie na internet.

Scenár strednej úrovne bezpečnosti AIX Security Expert

Ide o scenár pre strednú úroveň bezpečnosti AIX Security Expert.

Alica chce zlepšiť bezpečnosť systému, ktorý bude zapojený do podnikovej siete a ktorý sa nachádza za podnikovým firewallom. Sieť je bezpečná a dobre spravovaná. Tento systém bude používať veľký počet užívateľov, ktorí potrebujú prístup do **telnet** a **ftp** systému. Alica chce primerané nastavenia spoločnej bezpečnosti, napríklad ochranu skenovania portov a uplynutia doby platnosti hesiel, ale systém musí byť zároveň otvorený najvzdialenejším prístupovým metódam. V tomto scenári je najvhodnejším nastavením bezpečnosti pre Alicin systém stredná úroveň bezpečnosti.

Scenár nízkej úrovne bezpečnosti AIX Security Expert

Ide o scenár nízkej úrovne bezpečnosti AIX Security Expert.

Jozef nejaký čas spravoval systém. Systém sa nachádza na izolovanej zabezpečenej lokálnej sieti. Tento systém sa používa pre veľké množstvo ľudí a služieb. Jozef chce zvýšiť minimálnu úroveň bezpečnosti, ale nemôže prerušiť žiadnu formu prístupu do systému. Nízka úroveň bezpečnosti je pre Jozefov počítač správna.

Kontrolný zoznam bezpečnosti

Nasleduje kontrolný zoznam bezpečnostných akcií, ktoré majú byť vykonané na novo nainštalovanom alebo existujúcom systéme.

Hoci tento zoznam nie je úplným kontrolným zoznamom bezpečnosti, môžete ho použiť ako základ pre vytvorenie kontrolného zoznamu bezpečnosti pre vaše prostredie.

- Pri inštalácii nového systému nainštalujte systém AIX z bezpečného základného média. Počas inštalácie vykonajte nasledujúce procedúry:
  - Na servery neinštalujte softvér pracovnej plochy, ako napríklad, CDE, GNOME alebo KDE.
  - Nainštalujte požadované bezpečnostné opravy a všetky odporúčané opravy na úrovni údržby a technológie. Pozrite si na webových stránkach IBM System p eServer Support Fixes (http://www.ibm.com/support/fixcentral) najnovšie servisné bulletiny, bezpečnostné poradenstvo a informácie o opravách.
  - Po úvodnej inštalácii vytvorte zálohu systému a uložte ju na bezpečnom mieste.
- Vytvorte zoznamy pre riadenie prístupu k súborom a adresárom s obmedzeným prístupom.
- Zakážte nepotrebné užívateľské kontá a systémové kontá, ako napríklad daemon, bin, sys, adm, lp a uucp. Odstránenie kont sa neodporúča, pretože sa tým odstránia informácie o konte, napríklad názvy a ID užívateľov, ktoré môžu byť stále prepojené s údajmi v záložných kópiách systému. Ak sa vytvorí užívateľ s predtým odstráneným ID užívateľa a vykoná sa obnova systému zo záložnej kópie, nový užívateľ môže mať problémy s prístupom do obnoveného systému.
- Obvyklým spôsobom skontrolujte súbory /etc/inetd.conf, /etc/inittab, /etc/rc.nfs a /etc/rc.tcpip a odstráňte všetky nepotrebné démony a služby.
- Overte, či sú správne nastavené povolenia pre nasledujúce súbory:

```
-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
```

- Zakážte možnosť vzdialeného prihlásenia pre konto typu root. Konto typu root by malo umožňovať prihlásenie len z konzoly systému.
- Povoľte auditovanie systému. Viac informácií nájdete v časti “Prehľad auditu” na strane 127.
- Povoľte politiku riadenia prihlásení. Viac informácií nájdete v časti “Riadenie prihlásenia” na strane 33.
- Zakážte oprávnenia užívateľov na spúšťanie príkazu xhost. Viac informácií nájdete v časti “Zvládnutie starostí s X11 a CDE” na strane 39.
- Zamedzte neoprávneným zmenám premennej prostredia **PATH**. Viac informácií nájdete v časti “Premenná prostredia PATH” na strane 53.
- Zakážte funkcie telnet, rlogin a rsh. Viac informácií nájdete v časti “Zabezpečenie TCP/IP” na strane 195.
- Vytvorte ovládacie prvky pre kontá užívateľov. Viac informácií nájdete v časti “Riadenie užívateľských kont” na strane 51.
- Zaveďte prísnu politiku hesiel. Viac informácií nájdete v časti “Heslá” na strane 62.
- Vytvorte kvóty diskového priestoru pre kontá užívateľov. Viac informácií nájdete v časti “Obnova zo stavu po prekročení kvóty” na strane 73.
- Používanie príkazu **su** povoľte len administrátorským kontám. Monitorujte protokoly príkazu **su** v súbore /var/adm/sulog.
- Povoľte zamykanie obrazovky pri používaní okien X-Window.
- Obmedzte prístup k príkazom **cron** a **at** len na kontá, ktoré tento prístup potrebujú.
- Použite alias pre príkaz **ls** na zobrazenie skrytých súborov znakov v názve súboru.
- Použite alias pre príkaz **rm** na zabránenie náhodného vymazania súborov zo systému.
- Zakážte nepotrebné sieťové služby. Viac informácií nájdete v časti “Sieťové služby” na strane 203.
- Často vytvárajte zálohy systému a overujte ich integritu.
- Prihláste sa na odber do e-mailového distribučného zoznamu venovaného problematike zabezpečenia.

Súhrn bežných systémových služieb AIX

V nasledovnej tabuľke sú uvedené najznámejšie systémové služby v rámci systému AIX. Predtým, než začnete zabezpečovať váš systém, prezrite si túto tabuľku.

Pred zabezpečením si svojho systému zálohujte všetky svoje pôvodné konfiguračné súbory, najmä tieto:

- /etc/inetd.conf
- /etc/inittab
- /etc/rc.nfs
- /etc/rc.tcpip

Služba	Démon	Spúšťa sa v súbore	Funkcia	Poznámky
inetd/bootps	inetd	/etc/inetd.conf	služby bootp pre bezdiskové klientske zariadenia	• Potrebné pre Network Installation Management (NIM) a vzdialené zavedenie systémov • Pracuje súbežne so službou tftp. • Vo väčšine prípadov je vhodné túto službu vypnúť.
inetd/chargen	inetd	/etc/inetd.conf	generátor znakov (len na testovacie účely)	• Dostupné ako služba TCP a UDP. • Poskytuje možnosť Odmietnuť servisné útoky • Ak netestujete sieť, vypnite túto službu.
inetd/cmsd	inetd	/etc/inetd.conf	služba kalendára (využívaná prostredím CDE)	• Táto služba je spustená ako proces root, z čoho vyplýva možné ohrozenie bezpečnosti. • Ak túto službu nepotrebujete v rámci prostredia CDE, vypnite ju. • Túto službu vypnite na databázových serveroch.
inetd/comsat	inetd	/etc/inetd.conf	upozorňuje na došlú elektronickú poštu	• Táto služba je spustená ako proces root, z čoho vyplýva možné ohrozenie bezpečnosti. • Zriedkavo požadovaná • Vypnite túto službu.
inetd/daytime	inetd	/etc/inetd.conf	zastaraná služba času (len na testovacie účely)	• Táto služba je spustená ako proces root. • Dostupné ako služba TCP a UDP. • Poskytuje príležitosť pre útoky prostredníctvom príkazu PING zamerané na zlyhanie služieb (Denial of Service). • Táto služba je zastaraná a slúži len na testovacie účely. • Vypnite túto službu.
inetd/discard	inetd	/etc/inetd.conf	služba /dev/null (len na testovacie účely)	• Dostupné ako služba TCP a UDP. • Využíva sa pri útokoch zameraných na zlyhanie služieb (Denial of Service). • Táto služba je zastaraná a slúži len na testovacie účely. • Vypnite túto službu.

Služba	Démon	Spúšťa sa v súbore	Funkcia	Poznámky
inetd/dtspc	inetd	/etc/inetd.conf	riadenie podprocesu CDE	• Táto služba je automaticky spustená démonom **inetd** , ako odpoveď na požiadavku klienta CDE na spustenie procesu na hostiteľovi démona. Je zraniteľná voči útokom. • Túto službu vypnite na databázových serveroch bez prostredia CDE. • Prostredie CDE môže fungovať aj bez tejto služby. • Ak túto službu skutočne nepotrebujete, vypnite ju.
inetd/echo	inetd	etc/inetd.conf	služba echo (len na testovacie účely)	• Dostupné ako služba UDP a TCP. • Môže byť použitá pri útokoch zameraných na zlyhanie služieb a zahltenie (Denial of Service alebo Smurf). • Používa sa vykonanie volania echo na iného užívateľa, za účelom prieniku cez zariadenie firewall, alebo na zahltenie siete. • Vypnite túto službu.
inetd/exec	inetd	/etc/inetd.conf	služba vzdialeného spustenia	• Táto služba je spustená ako proces root. • Vyžaduje, aby ste zadali užívateľský ID a heslo, ktoré sa odovzdávajú nechránené • Táto služba je veľmi náchylná na odchytenie údajov. • Vypnite túto službu.
inetd/finger	inetd	/etc/inetd.conf	služba finger zobrazujúca informácie o užívateľoch	• Táto služba je spustená ako proces root. • Poskytuje informácie o systéme a užívateľoch. • Vypnite túto službu.
inetd/ftp	inetd	/etc/inetd.conf	služba ftp (file transfer protocol)	• Táto služba je spustená ako proces root. • ID užívateľa a heslo sa prenášajú nechránené, čo umožňuje odchytenie týchto údajov. • Vypnite túto službu a používajte bezpečné užívateľské prostredie (SSH) pre verejné domény.
inetd/imap2	inetd	/etc/inetd.conf	Internet Mail Access Protocol	• Uistite sa, že používate najnovšiu verziu tohto servera. • Potrebné len ak používate poštový server. V opačnom prípade túto službu vypnite. • ID užívateľa a heslo sa prenášajú nechránené.
inetd/klogin	inetd	/etc/inetd.conf	prihlásenie Kerberos	• Táto služba je zapnutá, ak lokalita používa autentifikáciu Kerberos.
inetd/kshell	inetd	/etc/inetd.conf	užívateľské prostredie Kerberos	• Táto služba je zapnutá, ak lokalita používa autentifikáciu Kerberos.

Služba	Démon	Spúšťa sa v súbore	Funkcia	Poznámky
inetd/login	inetd	/etc/inetd.conf	služba rlogin	• Náchylné na falšovanie adries IP a DNS. • Údaje vrátane ID užívateľov a hesiel sa prenášajú nechránené. • Táto služba je spustená ako proces root. • Namiesto tejto služby používajte bezpečné užívateľské prostredie (SSH).
inetd/netstat	inetd	/etc/inetd.conf	hlásenie aktuálneho stavu siete	• V prípade spustenia na vašom systéme môže poskytnúť hackerom informácie o sieti. • Vypnite túto službu.
inetd/ntalk	inetd	/etc/inetd.conf	umožňuje vzájomnú komunikáciu medzi užívateľmi	• Táto služba je spustená ako proces root. • Táto služba nie je potrebná v prípade výrobných alebo backendových serverov. • Ak túto službu skutočne nepotrebujete, vypnite ju.
inetd/pcnfsd	inetd	/etc/inetd.conf	súborové služby PC NFS	• Ak túto služby momentálne nepoužívate, vypnite ju. • Ak potrebujete podobnú službu, pouvažujte nad službou Samba. Démon pcnfs je totiž starší ako uvedenie špecifikácie SMB od spoločnosti Microsoft.
inetd/pop3	inetd	/etc/linetd.conf	Post Office Protocol	• ID užívateľov a heslá sú odosielané nechránené. • Potrebné len v prípade, ak váš systém predstavuje poštový server, a ak klienti používajú aplikácie podporujúce len POP3. • Ak klienti využívajú IMAP, použite radšej tento protokol, prípadne použite služby POP3. Táto služba obsahuje tunel SSL (Secure Socket Layer). • Ak neprevádzkujete poštový server alebo existujú klienti, ktorí vyžadujú služby POP, vypnite túto službu.
inetd/rexd	inetd	/etc/inetd.conf	vzdialené spustenie	• Táto služba je spustená ako proces root. • Spolupracuje s príkazom **on**. • Vypnite túto službu. • Používajte radšej **rsh** a **rshd**.
inetd/quotad	inetd	/etc/inetd.conf	hlásenia o súborových kvótach (pre klientov NFS)	• Potrebné len ak sú spustené súborové služby NFS. • Ak nie je táto služba potrebná na poskytovanie údajov pre príkaz **quota**, vypnite ju. • Ak túto služby potrebujete využívať, zabezpečte, aby boli aplikované najnovšie opravy.
inetd/rstatd	inetd	/etc/inetd.conf	Kernel Statistics Server	• Ak potrebujete monitorovať systémy, použite SNMP a túto službu vypnite. • Túto službu vyžaduje príkaz **rup**.

Služba	Démon	Spúšťa sa v súbore	Funkcia	Poznámky
inetd/rusersd	inetd	/etc/inetd.conf	informácie o prihlásenom užívateľovi	• Toto nie je dôležitá služba. Vypnite túto službu. • Táto služba je spustená ako proces root. • Poskytuje zoznam aktuálnych užívateľov na vašom systéme a partnerov s užívateľmi
inetd/rwalld	inetd	/etc/inetd.conf	píše všetkým užívateľom	• Táto služba je spustená ako proces root. • Ak má váš systém interaktívnych užívateľov, možno si budete musieť túto službu ponechať • Ak sú vašimi systémami výrobné alebo databázové servery, nie je potrebná • Vypnite túto službu.
inetd/shell	inetd	/etc/inetd.conf	služba rsh	• Ak je to možné, túto službu vypnite. Používajte radšej bezpečné užívateľské prostredie (SSH). • Ak musíte používať túto službu, použite TCP Wrapper na zamedzenie falšovania a obmedzenie rizík • Potrebná pre program na distribúciu softvéru**Xhier**
inetd/sprayd	inetd	/etc/inetd.conf	testovanie šírenia RPC	• Táto služba je spustená ako proces root. • Môže byť potrebná na diagnostiku problémov siete NFS • Ak neprevádzkujete NFS, vypnite túto službu.
inetd/systat	inetd	/etc/inted.conf	správa o stave "ps -ef"	• Umožňuje vzdialeným lokalitám získať informácie o stave procesov vo vašom systéme. • Táto služba je predvolene vypnutá. Je potrebné pravidelne kontrolovať jej stav, aby ste sa uistili, že táto služba nebola zapnutá.
inetd/talk	inetd	/etc/inetd.conf	umožňuje prostredníctvom rozdelenej obrazovky komunikovať dvom užívateľom v sieti	• Toto nie je požadovaná služba. • Používa sa spolu s príkazom **talk**. • Poskytuje službu UDP na porte 517. • Vypnite, pokiaľ nepotrebujete viaceré interaktívne chatovacie relácie pre užívateľa UNIX
inetd/ntalk	inetd	/etc/inetd.conf	umožňuje prostredníctvom príkazu "new talk" komunikovať dvom užívateľom v sieti	• Toto nie je požadovaná služba. • Používa sa spolu s príkazom **talk**. • Poskytuje službu UDP na porte 517. • Vypnite, pokiaľ nepotrebujete viaceré interaktívne chatovacie relácie pre užívateľa UNIX
inetd/telnet	inetd	/etc/inetd.conf	služba telnet	• Podporuje relácie vzdialeného prihlasovania, ale heslo a ID sa odovzdávajú nechránené • Ak je to možné, vypnite túto službu a pre vzdialený prístup používajte radšej bezpečné užívateľské rozhranie (SSH).

Služba	Démon	Spúšťa sa v súbore	Funkcia	Poznámky
inetd/tftp	inetd	/etc/inetd.conf	jednoduchý prenos súborov	• Poskytuje službu UDP na porte 69. • Táto služba je spustená ako proces root a môže preto predstavovať bezpečnostné riziko. • Túto službu využíva NIM. • Vypnite, pokiaľ nepoužívate NIM alebo pokiaľ nemusíte zaviesť operačný systém na pracovnú stanicu bez disku
inetd/time	inetd	/etc/inetd.conf	zastaraná časová služba	• Interná funkcia **inetd**, ktorú používa príkaz **rdate**. • Dostupné ako služba TCP a UDP. • Niekedy sa používa na synchronizáciu hodín pri zavádzaní operačného systému. • Táto služba je zastaraná. Namiesto nej použite ntpdate • Túto službu vypnite, len ak ste systém otestovali s vypnutou službou (zavedením a opätovným zavedením systému) a nezaznamenali ste pritom žiadne problémy.
inetd/ttdbserver	inetd	/etc/inetd.conf	databázový server tool-talk (pre CDE)	• **rpc.ttdbserverd** je spustené ako proces root a môže preto predstavovať bezpečnostné riziko. • Táto služba sa spúšťa ako služba požadovaná pre prostredie CDE, avšak CDE môže pracovať aj bez nej. • Nemala by byť spustená na backendových serveroch ani na systémoch, pre ktoré je dôležitá bezpečnosť.
inetd/uucp	inetd	/etc/inetd.conf	sieť UUCP	• Ak nepoužívate aplikáciu, ktorá využíva UUCP, túto službu vypnite.
inittab/dt	init	skript /etc/rc.dt v /etc/inittab	prihlásenie do pracovnej plochy prostredia CDE	• Spúšťa server X11 na konzole • Podporuje X11 Display Manager Control Protocol (xdcmp), aby sa ostatné stanice X11 mohli prihlásiť na ten istý počítač • Táto služba by mala byť používaná len na osobných pracovných staniciach. Nepoužívajte ju na backendových systémoch.
inittab/dt_nogb	init	/etc/inittab	prihlásenie do pracovnej plochy prostredia CDE (bez grafického zavedenia operačného systému)	• Žiadne grafické prostredie až do úplného zavedenia systému. • Rovnaké problémy ako pri inittab/dt
inittab/httpdlite	init	/etc/inittab	web server pre príkaz **docsearch**	• Predvolený web server pre mechanizmus docsearch. • Ak počítač nepredstavuje server dokumentácie, vypnite túto službu.

Služba	Démon	Spúšťa sa v súbore	Funkcia	Poznámky
inittab/i4ls	init	/etc/inittab	servery pre správu licencií	• Zapnite v prípade vývojových počítačov. • Vypnite v prípade výrobných počítačov. • Zapnite v prípade backendových databázových počítačov, ktoré majú licenčné požiadavky. • Poskytuje podporu pre kompilátory, databázový softvér a ďalšie licencované produkty.
inittab/imqss	init	/etc/inittab	vyhľadávací mechanizmus pre "docsearch"	• Súčasť predvoleného web servera pre mechanizmus docsearch. • Ak počítač nepredstavuje server dokumentácie, vypnite túto službu.
inittab/lpd	init	/etc/inittab	tlačové rozhranie pre BSD	• Prijíma tlačové úlohy z iných systémov. • Túto službu možno vypnúť a naďalej posielať úlohy tlačovému serveru. • Ak ste odskúšali, že tlač nie je vypnutím služby ovplyvnená, vypnite ju.
inittab/nfs	init	/etc/inittab	služby NFS a NIS (Network File System/Net Information Services)	• Služby NFS a NIS založené na UDP/RPC. • Autentifikácia je minimálna. • V prípade backendových počítačov vypnite tieto služby.
inittab/piobe	init	/etc/inittab	spracovanie vstupno-výstupných požiadaviek tlačiarne (pre tlač)	• Spracúva plánovanie, spoolovanie a tlač úloh odovzdaných démonom **qdaemon** • Ak tlačové úlohy odosielate na server a netlačíte priamo z vášho systému, vypnite túto službu.
inittab/qdaemon	init	/etc/inittab	démon frontu (pre tlač)	• Odosiela tlačové úlohy démonovi **piobe**. • Ak netlačíte priamo z vášho systému, vypnite túto službu.
inittab/uprintfd	init	/etc/inittab	správy jadra	• Táto služba väčšinou nie je potrebná. • Vypnite túto službu.
inittab/writesrv	init	/etc/inittab	zapisuje poznámky na terminály tty	• Používané len interaktívnymi užívateľmi pracovnej stanice UNIX • V prípade serverov, backendových databáz a vývojových počítačov túto službu vypnite. • Zapnite túto službu pre pracovné stanice.
inittab/xdm	init	/etc/inittab	tradičné grafické ovládanie X11	• Nespúšťajte na backendových výrobných ani databázových serveroch. • Nespúšťajte na vývojových systémoch, pokiaľ nie je potrebná správa obrazovky X11 • Ak sa požaduje grafické zobrazenie, možno spustiť na pracovných staniciach.
rc.nfs/automountd		/etc/rc.nfs	automatické súborové systémy	• Ak používate NFS, zapnite túto službu pre pracovné stanice. • Nepoužívajte v prípade vývojových alebo backendových serverov.
rc.nfs/biod		/etc/rc.nfs	Block IO démon (požadovaný pre server NFS)	• Zapnite len pre server NFS. • Ak nejde o server NFS, vypnite spolu s nfsd a rpc.mountd

Služba	Démon	Spúšťa sa v súbore	Funkcia	Poznámky
rc.nfs/keyserv		/etc/rc.nfs	Bezpečný server kľúčov RPC	• Spravuje kľúče potrebné pre bezpečné RPC. • Zakážte túto položku, ak *nepoužívate* NFS a NIS
rc.nfs/nfsd		/etc/rc.nfs	služby NFS (požadované pre server NFS)	• Autentifikácia je slabá. • Môže spôsobiť zlyhanie rámca zásobníka. • Zapnite v prípade súborových serverov NFS. • Ak túto službu vypnete, vypnite aj **biod**, **nfsd** a **rpc.mountd**.
rc.nfs/rpc.lockd		/etc/rc.nfs	uzamknutia súborov NFS	• Ak nepoužívate NFS, vypnite túto službu. • Ak v rámci siete nepoužívate uzamknutia súborov, vypnite túto službu. • Démon **lockd** je uvedený v zozname desiatich najväčších ohrození bezpečnosti (SANS Top Ten Security Threats).
rc.nfs/rpc.mountd		/etc/rc.nfs	pripojenia súborov NFS (požadovené pre server NFS)	• Autentifikácia je slabá. • Môže spôsobiť zlyhanie rámca zásobníka. • Zapnite v prípade súborových serverov NFS. • Ak túto službu vypnete, vypnite aj **biod** a **nfsd**.
rc.nfs/rpc.statd		/etc/rc.nfs	uzamknutia súborov NFS (na ich obnovenie)	• Implementuje uzamknutia súborov v rámci NFS. • Ak nepoužívate NFS, vypnite túto službu.
rc.nfs/rpc.yppasswdd		/etc/rc.nfs	démon hesiel NIS (pre nadradený systém NIS)	• Používa sa na úpravu lokálneho súboru hesiel. • Požadované len ak počítač predstavuje nadradený systém NIS, v opačnom prípade túto službu vypnite.
rc.nfs/ypupdated		/etc/rc.nfs	démon aktualizácie NIS (pre podradený systém NIS)	• Prijíma mapy databázy NIS dodané nadradeným systémom NIS. • Požadované len ak počítač predstavuje podradený systém NIS voči nadradenému serveru NIS.
rc.tcpip/autoconf6		/etc/rc.tcpip	rozhrania IPv6	• Vypnite, pokiaľ nespúšťate IP verziu 6
rc.tcpip/dhcpcd		/etc/rc.tcpip	protokol DHCP (Dynamic Host Configure Protocol) - klient	• Backendové servery by nemali používať DHCP. Vypnite túto službu. • Ak hostiteľ nevyužíva protokol DHCP, vypnite ju.
rc.tcpip/dhcprd		/etc/rc.tcpip	protokol DHCP (Dynamic Host Configure Protocol) - prenosový agent	• Prijíma DHCP vysielania a odosiela ich na server v ďalšej sieti. • Duplicitná služba sa nachádza v smerovačoch. • Ak nepoužívate protokol DHCP a nepotrebujete posielať informácie medzi sieťami, vypnite túto službu.

Služba	Démon	Spúšťa sa v súbore	Funkcia	Poznámky
rc.tcpip/dhcpsd		/etc/rc.tcpip	protokol DHCP (Dynamic Host Configure Protocol) - server	• Odpovedá na požiadavky DHCP od klientov v čase zavádzania systému a klientom poskytuje informácie ako názov IP, číslo, masku podiete, smerovač a adresu všemerového vysielania. • Ak nepoužívate protokol DHCP, vypnite túto službu. • V prípade výrobných a backendových serverov s hostiteľmi, ktorí nepoužívajú DHCP, túto službu vypnite.
rc.tcpip/dpid2		/etc/rc.tcpip	zastaraná služba SNMP	• Ak nepotrebujete používať SNMP, vypnite túto službu.
rc.tcpip/gated		/etc.rc.tcpip	bránové smerovanie medzi rozhraniami	• Emuluje funkcie smerovača. • Vypnite túto službu a používajte radšej protokol RIP alebo smerovač.
rc.tcpip/inetd		/etc/rc.tcpip	služby inetd	• Dôkladne zabezpečený systém by mal mať túto službu vypnutú, ale vypnutie tejto služby je často nepraktické. • Po vypnutí budú vypnuté aj služby vzdialeného užívateľského prostredia, ktoré vyžadujú niektoré poštové a web servery.
rc.tcpip/mrouted		/etc/rc.tcpip	smerovanie multi-cast	• Emuluje funkcie smerovača pre odosielanie paketov rozosielania (multi-cast) medzi segmentmi siete. • Vypnite túto službu. Používajte radšej smerovač.
rc.tcpip/names		/etc/rc.tcpip	server názvov DNS	• Túto službu používajte, len ak počítač slúži ako server názvov DNS. • V prípade pracovných staníc a vývojových a výrobných počítačov túto službu vypnite.
rc.tcpip/ndp-host		/etc/rc.tcpip	hostiteľ IPv6	• Vypnite, pokiaľ nepoužívate IP verziu 6
rc.tcpip/ndp-router		/etc/rc.tcpip	smerovanie IPv6	• Vypnite, pokiaľ nepoužívate IP verziu 6. Porozmýšľajte nad použitím smerovača namiesto IP verzie 6
rc.tcpip/portmap		/etc/rc.tcpip	služby RPC	• Požadovaná služba. • Servery RPC sa registrujú pomocou démona **portmap**. Klienti, ktorí potrebujú lokalizovať služby RPC zadajú požiadavku démonovi **portmap**, a ten im následne oznámi, kde sa konkrétna služba nachádza. • Túto službu vypnite, len ak ste zredukovali počet služieb RPC tak, že ste ponechali len službu **portmap**.
rc.tcpip/routed		/etc/rc.tcpip	smerovanie RIP medzi rozhraniami	• Emuluje funkcie smerovača. • Ak pre smerovanie paketov medzi sieťami máte k dispozícii smerovač, vypnite túto službu.
rc.tcpip/rwhod		/etc/rc.tcpip	vzdialený démon "who"	• Zhromažďuje a vysiela údaje partnerským serverom v tej istej sieti. • Vypnite túto službu.

Služba	Démon	Spúšťa sa v súbore	Funkcia	Poznámky
rc.tcpip/sendmail		/etc/rc.tcpip	poštové služby	• Táto služba je spustená ako proces root. • Ak počítač nepoužívate ako poštový server, vypnite túto službu. • Ak je vypnutá, urobte jeden z týchto krokov: – Aby sa vymazal front, zadajte položku do crontab. Použite príkaz **/usr/lib/sendmail -q**. – Nakonfigurujte služby DNS tak, aby bola pošta pre váš server doručovaná do iného systému.
rc.tcpip/snmpd		/etc/rc.tcpip	protokol SNMP (Simple Network Management Protocol)	• Ak systém nemonitorujete pomocou nástrojov SNMP, vypnite túto službu. • SNMP môže byť potrebný na kritických serveroch
rc.tcpip/syslogd		/etc/rc.tcpip	systémový protokol udalostí	• Vypnutie tejto služby sa *neodporúča* • Táto služba je náchylná voči útokom zameraným na zlyhanie služieb (Denial of Service). • Požadovaná v každom systéme.
rc.tcpip/timed		/etc/rc.tcpip	starý démon času	• Túto službu vypnite a používajte radšej xntp.
rc.tcpip/xntpd		/etc/rc.tcpip	nový démon času	• Má za úlohu synchronizáciu hodín v systémoch. • Vypnite túto službu. • Nakonfigurujte iné systémy ako časové servery a nechajte, aby sa ďalšie systémy synchronizovali podľa nich prostredníctvom príkazu cron, ktorý volá ntpdate.
dt login		/usr/dt/config/Xaccess	neobmedzené CDE	• Ak neposkytujete prihlásenie CDE pre skupinu staníc X11, môžete obmedziť prihlásenie dtlogin na konzolu.
anonymná služba FTP		user rmuser -p <username>	anonymné ftp	• Možnosť anonymného FTP zabraňuje sledovanie používania FTP konkrétneho užívateľa. • Nasledovným spôsobom odstráňte užívateľský ftp za predpokladu, že dané užívateľské konto existuje: **rmuser -p ftp** • Ďalšie zvýšenie zabezpečenia možno dosiahnuť zadaním zoznamu užívateľov do súboru /etc/ftpusers, ktorí nemajú mať prístup do systému pomocou služby ftp.

Služba	Démon	Spúšťa sa v súbore	Funkcia	Poznámky
anonymné zápisy FTP			anonymné odovzdania ftp	• Žiadny súbor by nemal patriť do ftp. • Anonymné odovzdania pomocou FTP predstavujú potenciálne riziko umiestnenia nebezpečného kódu do systému. • Mená užívateľov, ktorým chcete zamedziť prístup pomocou ftp zadajte do súboru /etc/ftpusers. • Nasleduje niekoľko príkladov systémom vytvorených užívateľov, ktorých anonymné odosielanie cez FTP na váš server budete možno chcieť zakázať: root, daemon, bin.sys, admin.uucp, guest, nobody, lpd, nuucp, ladp • Nasledovným spôsobom zmeňte práva vlastníka a skupiny súborom v adresári ftpusers: chown root:system /etc/ftpusers • Nasledovným spôsobom zmeňte oprávnenia súborom v adresári ftpusers a nastavte prísnejšie kritériá: chmod 644 /etc/ftpusers
ftp.restrict			prístup ftp k systémovým kontám	• Žiadnemu užívateľovi zvonka by nemalo byť povolené nahrádzať koreňové súbory pomocou súboru ftpusers
root.access		/etc/security/user	prístup rlogin/telnet ku kontu root	• Voľbu rlogin v súbore etc/security/user nastavte na false. • Každý užívateľ, ktorý sa chce prihlásiť ako užívateľ s oprávneniami typu root by sa mal najskôr prihlásiť pod vlastným menom a až potom prostredníctvom príkazu **su** ako užívateľ root. Tento postup umožňuje auditovanie.
snmpd.readWrite		/etc/snmpd.conf	skupiny SNMP readWrite	• Ak *nepoužívate* SNMP, vypnite démona SNMP. • Vypnite skupiny private a system v súbore /etc/snmpd.conf. • Skupinu 'public' obmedzte na tie adresy IP, ktoré monitorujú váš systém.
syslog.conf			configure syslogd	• Ak ste nenakonfigurovali /etc/syslog.conf, vypnite tohto démona. • Ak používate syslog.conf na protokolovanie systémových hlásení, ponechajte tohto démona zapnutého.

Súhrn volieb sieťových služieb

Na dosiahnutie vyššej úrovne zabezpečenia systému existuje niekoľko sieťových volieb, ktoré môžete zmeniť, a to vypnúť nastavením hodnoty 0 alebo zapnúť nastavením hodnoty 1. Nasledovný zoznam určuje parametre, ktoré môžete použiť s príkazom **no**.

Parameter	Príkaz	Účel
bcastping	/usr/sbin/no -o bcastping=0	Umožňuje odpovede na echo pakety ICMP na broadcast adresu. Vypnutím tejto voľby zabránite útokom zameraným na zahltenie (smurf).
clean_partial_conns	/usr/sbin/no -o clean_partial_conns=1	Určuje, či je nastavená ochrana pred útokmi typu SYN (synchronizuje poradové číslo).
directed_broadcast	/usr/sbin/no -o directed_broadcast=0	Určuje, či je povolené priame vysielanie na bránu. Nastavenie hodnoty 0 zabráni tomu, aby sa smerované pakety dostali do vzdialenej siete.
icmpaddressmask	/usr/sbin/no -o icmpaddressmask=0	Určuje, či systém odpovedá na požiadavky masky adresy ICMP. Vypnutím tejto voľby sa zabráni útokom typu source routing na získanie prístupu.
ipforwarding	/usr/sbin/no -o ipforwarding=0	Určuje, či má jadro postúpiť pakety. Vypnutím tejto voľby sa zabráni tomu, aby presmerované pakety dosiahli vzdialenú sieť.
ipignoreredirects	/usr/sbin/no -o ipignoreredirects=1	Určuje, či sa majú spracovať prijímané presmerovania.
ipsendredirects	/usr/sbin/no -o ipsendredirects=0	Určuje, či má jadro odosielať presmerované signály. Vypnutím tejto voľby sa zabráni tomu, aby presmerované pakety dosiahli vzdialenú sieť.
ip6srcrouteforward	/usr/sbin/no -o ip6srcrouteforward=0	Určuje, či systém postúpi source-routed pakety IPv6. Vypnutím tejto voľby sa zabráni útokom typu source routing na získanie prístupu.
ipsrcrouteforward	/usr/sbin/no -o ipsrcrouteforward=0	Určuje, či systém postúpi source-routed pakety. Vypnutím tejto voľby sa zabráni útokom typu source routing na získanie prístupu.
ipsrcrouterecv	/usr/sbin/no -o ipsrcrouterecv=0	Určuje, či systém akceptuje source-routed pakety. Vypnutím tejto voľby sa zabráni útokom typu source routing na získanie prístupu.
ipsrcroutesend	/usr/sbin/no -o ipsrcroutesend=0	Určuje, či aplikácie môžu odosielať source-routed pakety. Vypnutím tejto voľby sa zabráni útokom typu source routing na získanie prístupu.
nonlocsroute	/usr/sbin/no -o nonlocsrcroute=0	Oznamuje IP protokolu, že hostiteľovi mimo lokálnej siete je možné adresovať iba source-routed pakety. Vypnutím tejto voľby sa zabráni útokom typu source routing na získanie prístupu.
tcp_icmpsecure	/usr/sbin/no -o tcp_icmpsecurer=1	Chráni spojenia TCP pre útokmi typu ICMP (Internet Control Message Protocol) source quench a PMTUD (Path MTU Discovery). Skontroluje údajovú časť správy ICMP, aby sa otestovalo, či poradové číslo hlavičky TCP je v rozsahu akceptovateľných poradových čísel. Hodnoty: 0=vypnuté (štandardne); 1=zapnuté.
ip_nfrag	/usr/sbin/no -o ip_nfrag=200	Zadáva maximálny počet fragmentov paketu IP, ktorý môže naraz byť uchovávaný vo fronte znovuzostavenia IP (štandardná hodnota 200 uchováva maximálne 200 fragmentov paketu IP vo fronte znovuzostavenia IP).
tcp_pmtu_discover	/usr/sbin/no -o tcp_pmtu_discover=0	Vypnutím tejto voľby sa zabráni útokom typu source routing na získanie prístupu.

Parameter	Príkaz	Účel
tcp_tcpsecure	/usr/sbin/no -o tcp_tcpsecure=7	Chráni spojenia TCP pred zraniteľnosťou. Hodnoty: 0=žiadna ochrana; 1=posielanie falošného SYN do vytvoreného spojenia; 2=posielanie falošného RST do vytvoreného spojenia; 3=injektovanie údajov do vytvoreného spojenia TCP; 5–7=kombinácia vyššie uvedených zraniteľností.
udp_pmtu_discover	/usr/sbin/no -o udp_pmtu_discover=0	Vypne alebo zapne zisťovanie veľkosti paketov (path MTU discovery) pre aplikácie využívajúce protokol TCP. Vypnutím tejto voľby sa zabráni útokom typu source routing na získanie prístupu.

Viac informácií o nastaviteľných sieťových voľbách nájdete v príručke *Performance management*.

Dôveryhodný systém AIX

Dôveryhodný systém AIX povoľuje schopnosti viacúrovňovej bezpečnosti (MLS) v AIX.

Poznámka: MLS sa nazýva aj bezpečnosť založená na návestiach.

V porovnaní s bežným AIX, bezpečnosť Dôveryhodný systém AIX založená na návestiach implementuje návestia pre všetky subjekty a objekty v systéme.

Poznámka: Voľba inštalácie Dôveryhodný systém AIX povoľuje prostredie AIX s bezpečnosťou založenou na návestiach. Riadenie prístupov v systéme je založené na návestiach, ktoré poskytujú prostredie viacúrovňovej bezpečnosti (MLS) a zahŕňa podporu pre:

- Objekty s návestiami: súbory, objekty IPC, sieťové balíky a ostatné objekty s návestiami
- Tlačiarne s návestiami
- Dôveryhodnú sieť: podpora pre RIPSO a CIPSO v IPv4 a IPv6

Všimnite si, že po výbere tohto režimu inštalácie sa už nebudete môcť vrátiť do zvyčajného prostredia systému AIX bez vykonania inštalácie prepísaním bežného systému AIX. Skôr než si vyberiete tento režim inštalácie, vyhodnoťte svoju potrebu prostredia Dôveryhodný systém AIX. Podrobnosti o Dôveryhodný systém AIX nájdete vo verejne prístupnej dokumentácii systému AIX.

Štandardný systém AIX poskytuje súbor bezpečnostných funkcií, ktoré umožňujú manažérom a administrátorom informácií poskytovať základnú úroveň systémovej a sieťovej bezpečnosti. Primárne bezpečnostné funkcie systému AIX zahŕňajú:

- prístup do systému a siete riadený prihlásením a heslom
- užívateľské, skupinové a celosvetové oprávnenia na prístup do súborov
- zoznamy riadenia prístupov (ACL)
- podsystém auditu
- riadenie prístupu založené na rolách (RBAC)

Dôveryhodný systém AIX buduje na týchto primárnych bezpečnostných funkciách operačného systému AIX a ďalej zlepšuje a rozširuje bezpečnosť AIX do sieťových podsystémov.

Dôveryhodný systém AIX je kompatibilný s aplikačným programovým rozhraním (API) systému AIX. Každá aplikácia, ktorá používa systém AIX, môže používať aj Dôveryhodný systém AIX, avšak vzhľadom na ďalšie bezpečnostné obmedzenia sa môže stať, že aplikácie nepoznajúce MLS budú potrebovať na prevádzku v prostredí Dôveryhodný systém AIX privilégiá. Príkaz **tracepriv** sa používa na profilovanie aplikácií v takýchto scenároch.

Dôveryhodný systém AIX rozšíri API AIX na podporu ďalších bezpečnostných funkcií, čo umožní zákazníkom vyvinúť svoje vlastné bezpečnostné aplikácie pomocou rozhrania API AIX a nových rozšírení Dôveryhodný systém AIX.

Dôveryhodný systém AIX umožňuje systémom AIX spracúvať informácie na viacerých úrovniach bezpečnosti a je navrhnutý tak, aby spĺňal kritériá TCSEC Ministerstva obrany USA a európske kritériá ITSEC pre lepšiu bezpečnosť B1.

Informácie o štandardnej bezpečnosti systému AIX nájdete v kapitole Zabezpečenie základného operačného systému a Zabezpečenie siete.

Úvod do Dôveryhodný systém AIX

Dôveryhodný systém AIX rozširuje zabezpečenie štandardného operačného systému AIX poskytovaním funkcií bezpečnosti založenej na návestiach v operačnom systéme.

Prostredie založené na návestiach Dôveryhodný systém AIX môžete nainštalovať vybratím inštalačných volieb. Ak nainštalujete Dôveryhodný systém AIX, nebudete sa môcť vrátiť späť do bežného prostredia AIX, ak túto inštaláciu neprepíšete inštaláciou bežného AIX. Po nainštalovaní sa prostredie Dôveryhodný systém AIX použije na celý systém AIX vrátane všetkých WPAR vytvorených v prostredí AIX. Aj keď je zabezpečenie založené na návestiach (známe aj ako viac-úrovňové zabezpečenie alebo MLS) najčastejšie používané v odvetví bezpečnosti a tajných služieb, po upravení návestí v Dôveryhodný systém AIX môže byť použité aj komerčnými spoločnosťami. Nová inštalácia prostredia Dôveryhodný systém AIX poskytuje návestia, ktoré sú v súlade so štandardnými implementáciami MLS.

Prostredie Dôveryhodný systém AIX sa skladá z bežného AIX s niekoľkými ďalšími balíkmi a sadami súborov. Navyše, prepínače jadra prinútia jadro pracovať v režime Dôveryhodný systém AIX. Pri zavedení prostredníctvom CD alebo DVD sa systém bootuje v bežnom prostredí AIX. Keď sú zobrazené inštalačné ponuky, užívateľ, ktorý vykonáva inštaláciu, môže vybrať voľbu Dôveryhodný systém AIX a tým spustiť inštaláciu súborov pre MLS. Keď je inštalácia dokončená, užívateľ musí spustiť prvú zavádzaciu sekvenciu. Počas prvej zavádzacej sekvencie Config Assistant poskytne ponuky pre rôznych užívateľov a sú nastavení užívatelia ISSO, SA a SO, potom systém dokončí operáciu zavádzania a vytvorí MLS.

Dôveryhodný systém AIX rozširuje zabezpečenie systému cez štyri elementy zabezpečenia informácií:

- Dôvernosť
- Integritu
- Dostupnosť
- Sledovateľnosť

Okrem bezpečnostných funkcií, ktoré poskytuje AIX, systém Dôveryhodný systém AIX pridáva tieto schopnosti:

Návestia citlivosti (SL)
: Všetky procesy a súbory sú označené podľa príslušnej úrovne bezpečnosti. Procesy môžu vstupovať len do objektov, ktoré sú v rozsahu bezpečnosti daného procesu.

Návestia integrity (TL)
: Všetky procesy a súbory sú označené podľa príslušnej úrovne integrity. Súbory nemôžu byť zapísané procesmi, ktoré majú návestie nižšej úrovne integrity než súbor. Procesy nemôžu čítať zo súborov, ktoré majú návestie nižšej úrovne integrity než daný proces.

Bezpečnostné príznaky súboru
: Jednotlivé súbory môžu mať ďalšie príznaky na riadenie operácií týkajúcich sa bezpečnosti.

Bezpečnostné príznaky jadra
: Celý systém môže mať rôzne funkcie bezpečnosti zapnuté alebo vypnuté.

Privilégiá
: Mnohé volania príkazov a systémové volania sú dostupné len pre procesy s určitými privilégiami.

Oprávnenia
Každému užívateľovi môže byť pridelený jedinečný súbor oprávnení. Každé oprávnenie umožňuje užívateľovi vykonávať určité funkcie týkajúce sa bezpečnosti. Oprávnenia sú užívateľom priradené cez roly.

Roly Funkcia riadenia prístupov na základe rolí, ako súčasť prostredia Dôveryhodný systém AIX, zabezpečuje selektívne delegovanie administračných úloh na užívateľov. Toto delegovanie sa uskutočňuje zbieraním relevantných oprávnení do roly a priradením tejto roly užívateľovi.

Dôvernosť

Hrozba prezradenia informácií neoprávneným stranám sa považuje za problém dôvernosti.

Dôveryhodný systém AIX poskytuje mechanizmy opakovaného použitia objektu a riadenia prístupov na ochranu všetkých prostriedkov údajov. Operačný systém zabezpečuje, aby boli prostriedky chránených údajov prístupné len pre konkrétnych oprávnených užívateľov a aby ich títo užívatelia nemohli úmyselne či neúmyselne sprístupniť neoprávneným užívateľom.

Administrátori môžu predchádzať tomu, aby boli citlivé súbory zapísané na diskety alebo iné odstrániteľné médiá, vytlačené na nechránených tlačiarňach alebo presúvané cez sieť do neoprávnených vzdialených systémov. Túto ochranu bezpečnosti uplatňuje operačný systém a nie je možné, aby ju neoprávnení užívatelia alebo nebezpečné procesy obišli.

Integritu

Hrozba modifikácie informácií neoprávnenými stranami sa považuje za problém integrity.

Dôveryhodný systém AIX ponúka množstvo bezpečnostných mechanizmov, ktoré zabezpečujú integritu dôveryhodnej výpočtovej bázy a chránených údajov bez ohľadu na to, či sú údaje vygenerované v systéme alebo naimportované prostredníctvom sieťových prostriedkov. Rôzne bezpečnostné mechanizmy riadenia prístupov zabezpečujú, aby mohli informácie modifikovať len oprávnení užívatelia. Dôveryhodný systém AIX odstraňuje koreňové privilégium, aby tak zabránil neoprávneným užívateľom alebo nebezpečným procesom zmeniť veľkosť alebo vypnúť systémové procesy. Špeciálne administratívne oprávnenia a roly umožňujú namiesto udeľovania koreňových privilégií užívateľovi separáciu administračných povinností.

Dostupnosť

Hrozba týkajúca sa dostupnosti služieb na hostiteľskom počítači sa považujú za problém dostupnosti. Ak napríklad nebezpečný program vyplní priestor súboru tak, že nie je možné vytvoriť nový súbor, ešte stále existuje prístup, ale nie dostupnosť.

Dôveryhodný systém AIX chráni systém pred útokmi neoprávnených užívateľov a procesov, ktoré môžu spôsobiť odmietnutie služby. Neprivilegované procesy nemajú povolené čítať chránené súbory a adresáre ani do nich zapisovať.

Sledovateľnosť

Hrozba týkajúca sa poznania, ktoré procesy vykonali ktoré akcie v systéme, sa považuje za problém sledovateľnosti. Ak napríklad nie je možné sledovať užívateľa alebo proces, ktorý zmenil systémový súbor, nemôžete určiť, ako takéto akcie v budúcnosti zastaviť.

Táto vylepšená bezpečnostná funkcia zabezpečuje identifikáciu a autentifikáciu všetkých užívateľov ešte predtým, než povolí užívateľom prístup do systému. Služby auditu poskytujú administrátorovi súbor auditovateľných udalostí a stopu auditu všetkých udalostí týkajúcich sa bezpečnosti systému.

Vlastnosti Dôveryhodný systém AIX

- Dôveryhodný systém AIX sa nainštaluje cez ponuky inštalácie AIX. Ďalšie voľby môžete nastaviť počas inštalácie Dôveryhodný systém AIX.

- Prostredie Dôveryhodný systém AIX nie je možné prepnúť späť na bežné prostredie AIX, ak nevykonáte prepísanie inštaláciou bežného AIX.
- V prostredí Dôveryhodný systém AIX je protokolovanie pre užívateľa root vypnuté.
- V prostredí Dôveryhodný systém AIX, všetky vytvorené oddiely WPAR budú tiež fungovať v prostredí bezpečnosti na základe návestí.
- Dôveryhodný systém AIX podporuje MAC (Mandatory Access Control) a MIC (Mandatory Integrity Control). Zákazníci môžu definovať oddelené sady návestí pre MAC a MIC.
- Súbor šifier návestí je umiestnený v adresári /etc/security/enc a sú v ňom zaznamenané informácie o konverziách návestie-binárny formát. Štandardný súbor šifier návestí dodržiava požiadavky na pomenovanie pre Compartmented Mode Workstations (CMW) pre návestia.
- Inštalácie NIM sú podporované, ak sú spustené z klienta. Vynútenie inštalácie NIM zo servera nie je možné, pretože prihlasovanie užívateľa root je na systémoch MLS vypnuté.
- Súborový systém JFS2 (J2) (pomocou rozšírených atribútov verzie 2) bol povolený na ukladanie štítkov v AIX. Iné súborové systémy (ako J1 alebo NFS) môžu byť pripojené len v prostredí Dôveryhodný systém AIX ako jednoúrovňové súborové systémy (návestie je priradené k bodu pripojenia).
- Prostredie X je pre Dôveryhodný systém AIX vypnuté.
- Dôveryhodný systém AIX podporuje protokoly CIPSO a RIPSO pre sieťovú komunikáciu založenú na návestiach. Tieto protokoly sú podporované pre IPv4 aj IPv6.
- Niektoré bezpečnostné mechanizmy AIX sú v bežnom AIX a Dôveryhodný systém AIX spoločné. Dve z týchto spoločných bezpečnostných mechanizmov sú Role Based Access Control (RBAC) a Trusted Execution na overovanie integrity.
- Keďže je užívateľ root deaktivovaný, keď je nainštalované prostredie Dôveryhodný systém AIX, užívateľ, ktorý vykonáva inštaláciu, musí nastaviť heslá pre užívateľov ISSO, SA a SO počas prvého zavádzania po inštalácii. Systém bude v nepoužiteľnom stave, kým tieto heslá nebudú vytvorené.
- Publikácia Redbooks s názvom AIX 6 security features obsahuje prípady použitia a príklady pre Dôveryhodný systém AIX.

Viacúrovňová bezpečnosť

Hlavným cieľom bezpečného systému je zabezpečiť sledovateľnosť a dostupnosť tým, že uplatní bezpečnostnú politiku umiestnenia.

Bezpečnostná politika Dôveryhodný systém AIX poskytuje zadefinovanú skupinu pravidiel, ktoré určujú typy prípustných systémových prístupov. To zahŕňa udržať zodpovednosť užívateľov za ich zásahy a zamedziť zmenám v operačnom systéme.

Pri riadení prístupov k súborom, adresárom, procesom a zariadeniam využíva Dôveryhodný systém AIX riadenie prístupov a špecifické kritériá požadovaných informácií.

Dôveryhodný systém AIX udržiava záznam priebehu všetkých udalostí dôležitých z hľadiska bezpečnosti. Vďaka záznamu priebehu je možné sledovať zodpovednosť jednotlivo, dokonca aj pri programoch, ktoré upravujú pracovné a skutočné ID, ako napríklad príkaz **su**. Dôveryhodný systém AIX tiež obmedzuje administračné funkcie konkrétnych jednotlivcov s oprávneniami a najnižším privilégiom (udelenie najobmedzujúcejšej sady privilégií, ktorá užívateľovi, alebo procesu umožní vykonať operáciu).

Identifikácia a autentifikácia

Bezpečnostné mechanizmy identifikácia a autentifikácia (I&A) ručia za to, že každá osoba požadujúca prístup do systému, bude mať náležitú identifikáciu a autentifikáciu. Identifikácia vyžaduje meno užívateľa a autentifikácia vyžaduje heslo

Všetky kontá Dôveryhodný systém AIX sú chránené heslom. ISSO (Information Systems Security Officer) môže nakonfigurovať systém, aby umožňoval užívateľovi vybrať si svoje vlastné heslo, ktoré bude dodržiavať dĺžku hesla a obmedzenia pre zložitosť. ISSO tiež môže zadať parametre minimálneho a maximálneho veku hesiel (doby skončenia platnosti) jednotlivo pre každého užívateľa, vrátane dôb varovania pred uplynutím doby platnosti hesla.

Bezpečnostné mechanizmy identifikácia a autentifikácia vyžadujú, aby boli všetky mená užívateľov a ID užívateľov jedinečné. Kontá bez platných hesiel sa nemôžu použiť na prihlásenie. Užívateľ s rolou ISSO musí úvodné heslo pridať všetkým novým užívateľom. Každému užívateľovi bude priradený ďalší jedinečný identifikátor, ktorý sa používa na účely auditu.

Ukladá sa len šifrovaná forma hesla. Heslá nie sú v systéme uložené vo forme jednoduchého textu. Šifrované heslá sú uložené v tieňovom súbore hesiel, ktorý je chránený pred prístupom s výnimkou privilegovaných procesov. Bližšie informácie nájdete v príkaze **passwd**.

Systémy Dôveryhodný systém AIX rozoznávajú dva typy kont: systémové kontá a kontá užívateľov. Systémové kontá sú tie, ktoré majú ID užívateľa kratšie ako 128. Hoci môžu byť k systémovým kontám priradené heslá, nemôžu sa používať na prihlasovanie do systému.

Ľubovoľné riadenie prístupu

Ľubovoľné riadenie prístupu (DAC) sú aspekty bezpečnosti, ktoré sú riadené vlastníkom súboru, alebo adresára.

Oprávnenia UNIX

Užívateľ, ktorý má k prostriedku vlastnícke prístupové oprávnenie, môže urobiť nasledujúce:

- priamo prideliť prístup inému užívateľovi,
- prideliť inému užívateľovi prístup ku kópii,
- poskytnúť program, aby umožnil prístup k pôvodnému prostriedku (napríklad pomocou programov SUID)

Tradičná metóda bitov oprávnení UNIX (vlastník/skupina/iné a čítanie/zápis/spustenie) je príkladom tejto funkcionality DAC.

Bity oprávnení umožňujú užívateľom prideľovať, alebo odmietať užívateľom a skupinám prístup k údajom v súbore (v závislosti na kritériu požadovaných informácií). tento typ prístupu je založený na ID užívateľa a na skupinách, do ktorých tento užívateľ patrí. Všetky objekty súborového systému majú priradené oprávnenia, ktoré popisujú prístup pre vlastníka, skupinu a svet.

Vlastník súboru môže tiež prideliť prístupové privilégiá ostatným užívateľom, ak príkazmi **chown** a **chgrp** zmení vlastníctvo, alebo skupinu súboru.

Proces umask

Pri vytvorení súboru sú najprv všetky bity oprávnení zapnuté. Určité bity oprávnení sú potom súboru odňaté procesom umask, ktorý bol nastavený počas procesu prihlásenia. Štandardný proces umask je použitý na každý súbor vytvorený prostredím shell užívateľa a na každý príkaz, spustený v prostredí shell užívateľa.

Nastavenie umask pre položky je štandardne 000 (čo ponecháva všetky oprávnenia dostupné pre všetkých užívateľov). AIX nastaví umask jadra na 022 (čím vypne bity oprávnenia zápisu pre skupiny a pre svet). Užívatelia však v prípade potreby môžu tieto nastavenia prepísať.

Poznámka: Buďte veľmi opatrní pri zmenách umask na nastavenia povoľujúce viac, než 022. Ak sú pre súbory a procesy dostupné vyššie oprávnenia, stáva sa systém ako celok menej bezpečným.
Existujú dve metódy prepísania predvoleného nastavenia umask:

- Môžete zmeniť hodnoty umask vo vašich súboroch .profile, .login, alebo .chsrc. Tieto zmeny ovplyvnia každý súbor, ktorý bude vytvorený počas vašej relácie prihlásenia.
- Úrovne umask pre jednotlivé procesy môžete zmeniť príkazom **umask**. Po spustení príkazu **umask** budú všetky novo vytvorené súbory ovplyvnené novou hodnotou umask, až kým nedôjde k jednej z dvoch nasledujúcich udalostí:
  - spustíte znova príkaz **umask**,

    ALEBO
  - Ukončíte prostredie shell, v ktorom bol príkaz **umask** vydaný.

Ak príkaz **umask** spustíte bez akýchkoľvek argumentov, vráti príkaz **umask** aktuálne hodnoty umask vašej relácie.

Mali by ste prihlasovacej relácii povoliť dediť hodnotu umask jadra (022) tým, že nebudete špecifikovať umask vo svojich profiloch. Menej obmedzujúce hodnoty umask, než 022, by mali byť používané len s vysokou obozretnosťou.

Ak sú pre určité súbory potrebné dodatočné oprávnenia, mali by byť tieto oprávnenia nastavené po vytvorení tohto súboru s uvážlivým použitím príkazu **chmod**.

Zoznamy riadenia prístupov (ACL)

Okrem štandardných bitov oprávnení a hodnôt umask UNIX, systém AIX podporuje aj zoznamy riadenia prístupov (ACL).

Bity oprávnení UNIX riadia prístup len pre vlastníka súboru, jednu skupinu a každého v systéme. S ACL môže vlastník súboru určiť prístupové práva pre ďalších konkrétnych užívateľov a skupiny. Podobne ako bity oprávnení súvisia ACL s jednotlivými objektmi systému, ako napríklad súbor alebo adresár.

Bity oprávnení setuid a setgid

Bity oprávnení setuid a setgid (nastaviť ID užívateľa a nastaviť ID skupiny) umožňujú, aby bol súbor programu spustený s ID užívateľa alebo ID skupiny vlastníka súboru, namiesto ID užívateľa, alebo ID skupiny osoby, ktorá program spúšťa. To je možné vykonať nastavením bitov setuid a setgid, ktoré sú priradené tomuto súboru. To umožňuje vývoj chránených podsystémov, kde môžu užívatelia pristupovať k a spúšťať určité súbory bez toho, aby museli byť vlastníkmi týchto súborov.

Ak je bit setgid nastavený pri vytvorení objektu na jeho rodičovskom adresári, bude mať nový objekt rovnakú skupinu, ako rodičovský adresár, namiesto skupiny toho, kto tento objekt vytvoril. Objekty vytvorené v adresári s nastaveným bitom setuid, sú však vlastnené tvorcom objektu, nie vlastníkom adresára. Bity setuid/setgid rodičovského adresára sú dedené podadresármi, keď sú tieto podadresáre vytvorené.

Bity oprávnení setuid a setgid predstavujú potenciálne bezpečnostné riziko. Program, ktorý je nastavený, aby bol spúšťaný s užívateľom typu root ako vlastníkom by mohol mať v podstate neobmedzený prístup do systému. V systémoch Dôveryhodný systém AIX však toto bezpečnostné riziko značne znižuje využívanie privilégií a iného riadenia prístupov.

Elementy RBAC (Role Based Access Control)

Dôveryhodný systém AIX podporuje RBAC (Role Based Access Control). RBAC je mechanizmus operačného systému, prostredníctvom ktorého môžu byť systémové funkcie, ktoré sú špecifické pre užívateľa root/systémového superužívateľa, vykonávané aj obyčajnými užívateľmi použitím rolí, ktoré sú im priradené.

Základné elementy AIX RBAC sú:

Autorizácie
: Tieto reťazce indikujú operáciu privilégia, ktoré reprezentujú a riadia priamo podľa názvu. Napríklad, autorizačný reťazec aix.network.manage definuje funkciu správy siete v operačnom systéme AIX.

Privilégiá
: Privilégium je atribút procesu, ktorý procesu umožňuje obísť špecifické obmedzenia a limity systému. Privilégiá sú prepojené s procesom a typicky sa získavajú prostredníctvom vykonania privilegovaného príkazu.

Roly
: Elementy rolí v AIX RBAC umožňujú užívateľom kombinovať skupinu riadiacich funkcií v systéme a priradiť tieto funkcie na riadenie obyčajnému užívateľovi. Roly v AIX pozostávajú z kolekcie autorizácií (tieto môžu byť systémové autorizácie aj voliteľné autorizácie) a z ľubovoľných iných rolí (ako podrolí).

Bližšie informácie o riadení prístupu na základe rolí nájdete v RBAC.

Mandatory Access Control

Mandatory Access Control (MAC) je systémom vynútená metóda obmedzenia prístupu k objektom v závislosti na citlivosti objektu a na formálnych oprávneniach užívateľa. Protikladom je Discretionary Access Control (DAC), ktorá je viac, než systémom, vynútená jednotlivými vlastníkmi súborov.

Využívanie označení pri MAC

Pri vynútení MAC využíva Dôveryhodný systém AIX systém označení. Všetky pomenované objekty v systéme Dôveryhodný systém AIX majú označenie citlivosti (SL), ktorým je určená úroveň citlivosti každého objektu. Aj procesy majú označenie SL. SL procesu naznačuje, k akým úrovniam citlivých informácií má tento proces možnosť pristupovať. Všeobecne platí, že proces môže k objektu pristupovať vtedy, ak je úroveň citlivosti procesu rovnaká, alebo vyššia, než úroveň citlivosti objektu. Pomocou SL je možné zabezpečiť, aby boli súbory dostupné len na čítanie, alebo úplne zamedziť, aby k súborom mali prístup bežní užívatelia.

Všetky systémové objekty, akými sú súbory, IPC objekty, sieťové pripojenia a procesy, majú svoje SL. Pri vytváraní objektov sú SL na tieto objekty umiestnené automaticky. Za objekty sú považované aj všetky výpisy pamäte jadra, a preto sú systémom automaticky označené.

Objekty, ktoré existovali pred inštaláciou Dôveryhodný systém AIX dostanú štandardné označenie SYSTEM_LOW SL (SLSL), keď k nim systém pristupuje po inštalácii Dôveryhodný systém AIX. SL nie sú na týchto objektoch umiestnené natrvalo. Aby bolo nastavené SL, musí byť na objekte spustený príkaz **settxattr**. Objekty, ktoré sú vytvorené až po inštalácii Dôveryhodný systém AIX, majú označenie SL nastavené na SL procesu, ktorý ich vytvoril.

Užívatelia a označenia

Každému kontu užívateľa priradí systém rozsah platných SL, či už predvolených systémom, alebo podľa špecifických nastavení užívateľa a užívateľ môže fungovať len v rámci tohto rozsahu. Proces alebo užívateľ môže vytvárať súbory a adresáre len podľa aktuálneho označenia citlivosti procesu, alebo užívateľa a môže čítať a zapisovať do súborov len podľa systémom daných obmedzení MAC.

Uplatnenie metódy MAC

Metóda Mandatory Access Control je uplatnená zakaždým, keď proces vykoná pokus o otvorenie objektu súborového systému, získanie atribútov objektu súborového systému, odoslanie signálu procesu, prenos údajov cez STREAM, alebo odoslanie a prijatie paketu prostredníctvom sieťového rozhrania. Prístup ku ktorémukoľvek objektu súborového systému je možný len vtedy, keď sú splnené kritéria metód MAC aj DAC. Keď sa užívateľ pokúsi o prístup k súboru, sú obmedzenia MAC uplatnené skôr, než obmedzenia DAC, skontrolované sú napríklad bity oprávnení alebo zoznamy ACL.

Prístup k objektom súborového systému obmedzuje nie len SL objektu, ale aj SL adresára, v ktorom je objekt umiestnený. Preto môže byť objekt súborového systému chránený odlišnou úrovňou citlivosti (SL adresára), než je SL samotného objektu. Objekty súborových systémov môžu mať viac názvov (odkazov) umiestnených v jednom, alebo viacerých adresároch. Hoci je každý názov (odkaz) chránený rovnakým SL ako súbor, na ktorý je odkaz nasmerovaný, môže byť skutočná ochrana viacerých odkazov rozličná, keďže sú tieto odkazy v adresároch s rozličnou úrovňou ochrany.

Názov objektu je uložený v adresári, v ktorom je objekt umiestnený. Preto každý proces, ktorý má do adresára prístup, môže vidieť aj názvy objektov v tomto adresári. Čítať a zapisovať do týchto objektov však môžu len tie procesy, ktoré k nim majú zodpovedajúci prístup.

Vypísanie zoznamu SL a zmeny SL

Označenia SL objektov a procesov v systéme môžete zobraziť príkazom **lstxattr** a upravovať pomocou príkazov **settxattr**.

Zmeniť SL súboru, alebo procesu môžu len užívatelia, ktorí majú patričné oprávnenia a procesy s náležitými privilégiami.

Pri zmene SL objektu súborového systému na SL nižšej úrovne pomocou **settxattr** by užívateľ mal mať autorizáciu aix.mls.label.sl.downgrade. Ak chce užívateľ zvýšiť SL objektu súborového systému, mal by mať autorizáciu aix.mls.label.sl.upgrade. Pri zmenách SL procesov by mal mať užívateľ autorizáciu aix.mls.proc.sl.upgrade na ich zvýšenie autorizáciu aix.mls.proc.sl.downgrade na ich zníženie.

Metóda MAC na deskriptoroch otvárania súborov

Pri čítaní/zápise a jednoduchom prístupe k súborom sú kontroly MAC vykonávané, keď proces k súboru pristupuje. Keď má proces deskriptor súboru pre niektorý súbor, môže súbor čítať a zapisovať doň, aj keď je SL procesu zmenené na nižšiu úroveň, než je SL súboru. Niektoré operácie, napríklad nastavenie vlastníka, oprávnení, označení a privilégií súboru, však vykonávajú kontrolu prístupu aj po tom, ako proces získal deskriptor súboru.

To znamená, že keď proces pristupuje k súboru pomocou deskriptora súboru, nie sú vykonávané kontroly MAC a rozlíšenia cesty adresárov s oddielmi. Označenie SL súboru a/alebo procesu môže byť zmenené a prístup je stále povolený.

Mandatory Integrity Control

Mandatory Access Control (MAC) je systémom vynútená metóda obmedzenia prístupu a modifikácií objektov v závislosti na integrite objektu a na formálnych oprávneniach užívateľa. Zatiaľ čo metóda MAC sa zaoberá citlivosťou objektu, metóda MIC je zameraná na dôveryhodnosť objektu.

Využívanie označení pri MIC

Pri vynútení MIC využíva Dôveryhodný systém AIX systém označení. Všetky pomenované objekty v systéme Dôveryhodný systém AIX majú označenie integrity (TL), ktorým je určená úroveň integrity každého objektu. Aj procesy majú označenie TL. TL procesov naznačujú úroveň integrity informácií, k akým môže proces pristupovať. Čím je TL vyššie, tým je proces, alebo objekt dôveryhodnejší.

Aby proces mohol objekt modifikovať, musí byť minimálne tak dôveryhodný, ako tento objekt. Z toho dôvodu musí byť TL procesu rovnaké, alebo vyššie, než je TL objektu. Preto je pomocou označení integrity možné nastaviť súbory tak, aby boli dostupné len na čítanie.

Navyše proces nesmie použiť údaje z objektu, ktorý je menej dôveryhodný, než samotný proces. Z toho dôvodu musí byť TL objektu rovnaké, alebo vyššie, než TL procesu.

Všetky systémové objekty, akými sú súbory a procesy, majú označenie TL. TL sú na objekty umiestnené automaticky pri vytváraní objektov. Za objekty sú považované aj všetky výpisy pamäte jadra, a preto sú systémom automaticky označené.

Objekty, ktoré v systéme existovali pred inštaláciou Dôveryhodný systém AIX dostanú štandardné označenie SYSTEM_LOW TL (SLTL), keď k nim systém pristupuje po inštalácii Dôveryhodný systém AIX. TL nie sú na týchto objektoch umiestnené natrvalo. Aby bolo TL nastavené, musí byť na objekte spustený príkaz **settxattr**. Objekty, ktoré sú vytvorené až po inštalácii Dôveryhodný systém AIX, majú označenie TL nastavené na úroveň integrity procesu, ktorý ich vytvoril.

Užívatelia a označenia

Každému kontu užívateľa priradí systém rozsah platných TL, či už predvolených systémom, alebo podľa špecifických nastavení užívateľa a užívateľ môže fungovať len v rámci tohto rozsahu. Proces alebo užívateľ môže vytvárať súbory a adresáre len podľa aktuálneho označenia TL, alebo užívateľa a môže čítať a zapisovať do súborov len podľa systémom daných obmedzení MIC.

Uplatnenie metódy MIC

Metóda Mandatory Integrity Control je uplatnená zakaždým, keď je uplatnená metóda MAC. Ďalej je metóda MIC vynútená, keď je súbor, alebo adresár vymazaný, alebo premenovaný.

Zmena označení TL

Označenia TL objektov a procesov môžete zobraziť príkazom **lstxattr** a upraviť príkazom **settxattr**.

Zmeniť TL súboru, alebo procesu môžu len užívatelia, ktorí majú patričné oprávnenia a procesy s náležitými privilégiami. Pri zmene TL objektu súborového systému na TL nižšej úrovne pomocou **settxattr** by užívateľ mal mať autorizáciu aix.mls.label.tl.downgrade. Ak chce užívateľ zvýšiť TL objektu súborového systému, mal by mať autorizáciu aix.mls.label.tl.upgrade. Pri zmenách TL procesov by mal mať užívateľ autorizáciu aix.mls.proc.tl.upgrade na ich zvýšenie autorizáciu aix.mls.proc.tl.downgrade na ich zníženie.

NOTL

Existuje špeciálne TL, NOTL, ktoré je možné použiť na súborové systémy, objekty ipc, alebo procesy. Ak TL objektu, alebo procesu je NOTL, nie sú na tomto objekte, alebo procese vykonávané žiadne kontroly MIC. Nastaviť TL na NOTL, alebo zmeniť TL, ak aktuálne TL je NOTL, môže len privilegovaný užívateľ.

Metóda MIC na deskriptoroch otvárania súborov

Pri čítaní/zápise a jednoduchom prístupe k súborom sú kontroly MIC vykonávané, keď proces k súboru pristupuje. Keď má proces deskriptor súboru pre niektorý súbor, môže súbor čítať a zapisovať doň, aj keď je TL procesu zmenené na nižšiu úroveň, než je TL súboru. Niektoré operácie, napríklad nastavenie vlastníka, oprávnení, označení a privilégií súboru, však vykonávajú kontrolu prístupu aj po tom, ako proces získal deskriptor súboru. To znamená, že keď proces pristupuje k súboru pomocou deskriptora súboru, nie sú vykonávané kontroly MIC. Označenie TL súboru a/alebo procesu môže byť zmenené a prístup bude stále povolený.

Návestia

Návestia sa používajú, aby zastupovali bezpečnostné úrovne subjektov a objektov v systémoch Dôveryhodný systém AIX. Návestia, ktoré sa budú používať v systéme a vzťahy medzi návestiami definuje ISSO.

Návestia citlivosti (SL):

SL, priradené ku každému subjektu a objektu, sa používajú na presadenie povinnej politiky riadenia prístupov, založenej na modeli riadenie prístupov Bell-LaPadula.

SL pozostáva z dvoch častí:

- Hierarchická klasifikácia
- Skupina jedného alebo viacerých oddielov

Každá lokalita inštalácie môže definovať názvy a vzťahy návestí na danom systéme. Administrátor systému môže tieto názvy a vzťahy nastaviť podľa požiadaviek politík lokality v súbore kódovania návestí.

Členenie SL:

Členenia majú hierarchické usporiadanie a predstavujú úroveň citlivosti.

Napríklad, ak Top Secret, Secret a Unclassified sú v lokalite platnými členeniami, potom Top Secret je citlivejšie ako Secret a Secret je citlivejšie ako Unclassified. Dôveryhodný systém AIX podporuje maximálne 32 000 hierarchických členení.

Oddiely SL:

Oddiely predstavujú témy alebo pracovné skupiny. Každý oddiel má názov, ako napríklad NATO alebo CRYPTO.

Oddiely nemajú žiadne vnútorné usporiadanie, ale ISSO môže zaviesť obmedzenia pre kombinovanie oddielov s členeniami. Dôveryhodný systém AIX podporuje maximálne 1 024 oddielov.

Komponenty SL:

V tvare čitateľnom človekom je SL reprezentovaný reťazcom elementov. Prvý element predstavuje klasifikáciu; ostatné elementy predstavujú oddiely. Elementy sú oddelené medzerou.

Napríklad ak súbor obsahuje prísne tajné informácie ohľadom brazílskej ekonomiky, hierarchická klasifikácia súboru by mohla byť prísne tajný (TS) a oddiely by mohli byť Brazília (B) a ekonomika (e). Človekom čitateľný tvar SL by bol TS B e alebo Prísne tajný Brazília ekonomika.

Vzťahy SL:

Pre systémového užívateľa je dôležité chápať vzťahy medzi návestiami a spôsobom použitia návestí.

Existujú tri typy vzťahov medzi návestiami MAC:
- Nadradenosť
- Rovnosť
- Neporovnateľné

Nadradenosť

Jeden SL (L1) je nadradený druhému (L2) iba ak sú splnené obidve nasledujúce podmienky:
- Klasifikácia v L1 je rovná alebo vyššia ako klasifikácia v L2
- Sada oddielov v L1 obsahuje úplnú skupinu oddielov v L2

Napríklad ak predpokladáme jeden SL L1 prísne tajných informácií v oddieloch A a B (TS A B) a ďalší SL L2 tajných informácií v oddiele A ale nie v B (S A), potom TS A B by bol nadradený S A, lebo klasifikácia TS je nadradená klasifikácii S a skupina oddielov v L1 obsahuje úplnú skupinu oddielov v L2. V tomto príklade by L2 nebol nadradený L1.

Tabuľka 34. Nadradenosť SL

L1		L2		Nadradenosť
Návestie	**Oddiel**	**Návestie**	**Oddiel**	
TOP SECRET	A,B	SECRET	A	L1 > L2

Rovnosť

Jeden SL (L1) je považovaný za rovný inému SL (L2) iba ak sú splnené obidve nasledujúce podmienky:
- Klasifikácia v L1 je rovná klasifikácii v L2
- Sada oddielov v L1 je rovnaká ako skupina oddielov v L2

Ak sa dve návestia rovnajú, potom každé návestie je nadradené tomu druhému. Napríklad ak predpokladáme SL pre súbor s prísne tajnými informáciami v oddiele A (TS A) a iný súbor s prísne tajnými informáciami v oddiele A (taktiež TS A), potom SL by boli rovné a navzájom by si boli nadradené.

Tabuľka 35. Rovnosť SL

L1		L2		Nadradenosť
Návestie	**Oddiel**	**Návestie**	**Oddiel**	
TOP SECRET	A	TOP SECRET	A	L1 = L2

Neporovnateľné

Dva SL môžu byť disjunktívne (L1 nie je rovný L2, L1 nie je nadradený L2 a L2 nie je nadradený L1). jeden SL (L1) je považovaný za neporovnateľný s druhým (L2) iba ak sú splnené obidve nasledujúce podmienky:

- Sada oddielov v L1 neobsahuje úplnú skupinu oddielov v L2 a L2 neobsahuje úplnú skupinu v L1. Preto L1 a L2 sú považované za disjunktívne

Napríklad ak predpokladáme súbor s návestím L1 s prísne tajnými informáciami v oddieloch A a B (TS A B) a L2 je návestie pre súbor s dôvernými informáciami v oddiele C (C C), potom L1 je neporovnateľný s L2.

Tabuľka 36. Neporovnateľné SL

L1		L2		Nadradenosť
Návestie	**Oddiel**	**Návestie**	**Oddiel**	
TOP SECRET	A, B	CLASSIFIED	C	-

Návestia integrity (TL):

TL predstavujú úroveň dôveryhodnosti v systémovom objekte alebo v procese. TL majú rovnakú štruktúru ako SL, s jednou výnimkou, a síce TL majú len hierarchické členenie a nemajú žiadne oddiely.

Proces môže modifikovať alebo vymazať objekt len vtedy, ak je TL procesu nadradené nad TL objektu. Proces môže vymazať alebo premenovať objekt len vtedy, ak je TL procesu nadradené aj TL objektu aj TL adresára, v ktorom je objekt trvalo umiestnený. Proces môže pristúpiť na objekt len vtedy, ak je TL objektu nadradené TL procesu.

Ak chcete určiť TL objektu alebo procesu, použite príkaz **lstxattr**. Ak chcete zmeniť TL objektu alebo procesu, použite príkaz **settxattr**.

Návestia na subjektoch a na objektoch:

V Dôveryhodný systém AIX sú procesy identifikované ako subjekty a každý proces má niekoľko SL.

SL použité pre kontroly MAC sa nazýva Efektívne SL (ESL). ESL sa musí nachádzať v rozsahoch previerok procesov. Rozsah previerok má hornú hranicu a dolnú hranicu. Horná hranica sa nazýva Maximálna previerka (Max CL) a dolná hranica sa nazýva Minimálna previerka (Min CL). ESL, Max CL a Min CL sú uložené v štruktúre splnomocnení procesov a sú priraďované počas vytvárania procesu. Max CL musí byť nadradené Min CL a ESL a ESL musí byť nadradené Min CL. Na vypísanie zoznamu a nastavenie SL procesov sa používajú príkazy **settxattr** a **lstxattr**.

Prístup k rozličným objektom systému musí byť riadený. Objektom môžu byť:

- procesy
- súbory (údajové alebo binárne)
- objekty IPC, sieťové pakety, a pod.

Všetky objekty a subjekty na systéme MLS majú návestie.

Adresár

Adresárom je priradený rozsah SL od minimálneho SL až po maximálne SL. Maximálne SL by malo byť väčšie alebo rovné minimálnemu SL. Všetky súbory v adresári sú v tomto rozsahu.

Súbory
Štandardným súborom sú priradené dve SL, ale ich hodnoty sú vždy rovnaké. Takže prakticky majú len jedno SL. Symbolické odkazy môžu mať odlišné hodnoty pre tieto SL.

Špeciálne súbory
Špeciálnym súborom, ako sú zariadenia, tty a fronty fifo, je priradené maximálne a minimálne SL. Adresáre, súbory a špeciálne súbory majú len jedno návestie integrity (TL), kým procesom je priradené minimálne a maximálne TL.

Procesy
Všetkým procesom je priradený maximálny a minimálny rozsah povolení citlivosti, ako aj maximálny a minimálny rozsah povolení integrity. Tieto hodnoty sú zdedené od povolení užívateľa. Úrovne citlivosti a integrity, na ktorých je proces vykonávaný, sú úrovne efektívnej citlivosti a integrity.

Návestia previerok užívateľov:

Užívatelia majú návestia previerky maximálnej a minimálnej citlivosti (SCL) a návestia previerky maximálnej a minimálnej integrity (TCL)

Návestia previerky maximálnej a minimálnej citlivosti

Každý užívateľ má návestie previerky maximálnej citlivosti (max SCL). Efektívnemu SL užívateľa musí byť nadradené max SCL. max SCL sa používa na zamedzenie zobrazovania veľmi citlivých údajov určitými užívateľmi. min SCL sa používa na zamedzenie prenosu údajov od užívateľov na vysokej úrovni zabezpečenia k užívateľom na nižšej úrovni zabezpečenia.

Napríklad predpokladajme, že užívateľ A má aj max SCL aj min SCL s hodnotu PUBLIC_A a užívateľ B má aj max SCL aj min SCL s hodnotou PUBLIC_B. Bez min SCL by mohol užívateľ A oznamovať informácie užívateľovi B tak, že by sa prihlásil pomocou efektívneho SL s označením IMPL_LO a zapisoval by do súboru, ktorý by si potom užívateľ Bprečítal. S min SCL sa však užívateľ A musí prihlásiť na PUBLIC_A a súbory môže zapisovať len do PUBLIC_A. Žiadne súbory, zapísané do PUBLIC_A nemôže čítať užívateľ B.

Návestia previerky maximálnej a minimálnej integrity

Každý užívateľ má aj návestie previerky maximálnej integrity (max TCL). Efektívnemu TL užívateľa musí byť nadradené max TCL. max TCL sa používa na obmedzenie zobrazovania veľmi citlivých údajov určitým užívateľom. min TCL sa tiež používa na zamedzenie prenosu údajov od užívateľov na vysokej úrovni zabezpečenia k užívateľom na nižšej úrovni zabezpečenia.

Označenia na objektoch súborového systému:

Všetky súbory obsahujú špecifické bezpečnostné informácie. Keď je vytvorený nový súbor, má rovnaké označenie SL, ako proces, ktorý ho vytvoril. Označenie SL informácií v súbore je možné zvýšiť, alebo znížiť, a to zvýšením, alebo znížením SL súboru.

Adresárom je pri ich vytvorení priradené minimálne a maximálne označenie SL. Pri vytvorení sú oba nastavené rovnako, ako efektívne SL procesu, ktorý ich vytvoril, čím je v zásade vytvorený jednoúrovňový adresár. Tieto označenia SL môžu zmeniť len užívatelia s príslušnými privilégiami a autorizáciami. Nové objekty môžu byť v tomto adresári vytvorené len ak efektívne SL procesu vytvárajúceho nový objekt spadá do rozsahu označení SL adresára.

Okno je normálne vytvorené ako osobitný proces potomka s rovnakým označením SL, ako efektívne SL užívateľa. Zariadenia (napríklad okná priradené pseudoterminálom) majú tiež priradené označenia SL. Pomenovaný dátovod, ktorý je zariadením určeným na medziprocesovú komunikáciu, dedí efektívne SL procesu, ktorý tento pomenovaný dátovod vytvoril. Tok, ktorý je zariadením používaným na poskytnutie dvojsmerného údajového kanála pre medziprocesovú komunikáciu, tiež dedí efektívne SL procesu, ktorý tento tok vytvoril.

Všetky zariadenia majú minimálne SL a maximálne SL. Maximálne SL nesmie byť nižšie, než minimálne SL. Štandardne sú minimálne SL a maximálne SL nastavené rovnako. Proces môže pristupovať k takým zariadeniam v režime čítania len vtedy, ak SL procesu nie je menšie, než minimálne SL zariadenia, alebo adresára. Proces môže k takému zariadeniu pristupovať v režime zápisu len vtedy, ak je SL procesu v rámci rozsahu určenom minimálnym a maximálnym SL zariadenia, alebo adresára.

Bezpečnostné príznaky súboru

Objekty môžu byť označené bezpečnostnými príznakmi súborov (FSF), ktoré ovplyvňujú spôsob, akým proces s objektmi zaobchádza. V dokumente Bezpečnostné príznaky súborov nájdete zoznam FSF a privilégiá, ktoré sú pri nastavení každého FSF nevyhnutné. Procesy nemajú žiadne bezpečnostné príznaky súborov.

Odstraňovanie súborov:

Odstrániť objekt zo súborového systému môžete len vtedy, ak platí toto:

- Proces, ktorý sa pokúša odstrániť objekt, musí byť schopný vidieť názov súboru v adresári, ktorý obsahuje daný súbor. To znamená, že proces musí mať oprávnenie na vyhľadávanie v každom adresári na ceste smerom nadol do adresára, z ktorého má byť odstránený daný objekt a proces musí mať efektívny SL, ktorý je nadradený každému z týchto adresárov. Na zobrazenie názvu súboru použite príkaz **ls**.
- Proces musí mať oprávnenie na zápis na adresár, z ktorého má byť daný objekt odstránený.

Tlačové súbory:

Podsystém tlačiarne automaticky označuje všetky výstupy príslušnými označeniami citlivosti. Každá tlačová úloha je automatický opatrená stránkou s banerom a stránkou s trailerom, v ktorých sú zobrazené všetky podstatné bezpečnostné značky a označenia.

Zálohovanie a obnova súborov:

Pri zapisovaní na disky alebo pásky v systéme AIX pomocou príkazu **backup** sú do údajov zahrnuté návestia SL.

Na použitie príkazov **backup** a **restore** na importovanie alebo exportovanie údajov bez návestia z pások alebo diskov je vyžadované oprávnenie SO. Pri zapisovaní údajov bez návestia je údajom pridelené predvolené SL SYSTEM_LOW pre súbory a rozsah SL SYSTEM_LOW až SYSTEM_HIGH pre adresáre.

Návestia na objektoch IPC:

Všetky IPC zariadenia AIX sa zapájajú do vytvárania a pristupovania na prechodné objekty.

V AIX sú definované tri rôzne IPC zariadenia:

- Fronty správ
- Semafory
- Zdieľaná pamäť

Všetky z nich sa zapájajú do vytvárania a pristupovania na prechodné objekty, nazývané IPC objekty pre medziprocesovú komunikáciu. Každý IPC objekt je chránený sadou atribútov, ktoré sa podobajú na atribúty, ktoré chránia súbory. K týmto atribútom patrí:

- ID užívateľa a ID skupiny vlastníka objektu
- ID užívateľa a ID skupiny tvorcu objektu
- Režim prístupu k prostriedkom, ktorý sa zhoduje s bitmi oprávnení pre prístup k súborom. Každý objekt má oprávnenie na čítanie, zápis a spustenie pre svet, skupinu a vlastníka objektu.
- Poradové číslo pre sledovanie využívania prostriedkov
- Kľúč pre identifikáciu prostriedkov

Dôveryhodný systém AIX ako aj pri iných systémových objektoch rozširuje tieto atribúty o ďalšie bezpečnostné atribúty. V systéme Dôveryhodný systém AIX majú všetky IPC objekty aj nasledujúce atribúty:

- Návestie citlivosti (SL)
- Návestie integrity (TL)

Na zobrazenie všetkých bezpečnostných atribútov IPC objektu môžete použiť príkaz **settxattr**. Načítanie atribútov IPC objektu si vyžaduje oprávnenia DAC READ a MAC READ pre objekt.

Prístup k objektom IPC:

Objekty IPC sa vytvárajú, odstraňujú a pristupuje sa k nim prostredníctvom rôznych systémových volaní, ktoré sú popísané v téme Programovanie pre Dôveryhodný systém AIX. Bežní užívatelia tieto operácie nevykonávajú. Táto téma je všeobecným prehľadom pravidiel pre vytváranie, vymazávanie a prístup k objektom IPC.

Aby bol možný prístup k objektu IPC, proces musí prejsť kontrolami prístupu DAC, MIC a MAC.

Kontroly prístupu DAC sú založené na režime (vlastník, skupina alebo svet) objektu a ID užívateľa a skupiny procesu. Proces má prístup vlastníka DAC k objektu IPC, ak efektívne UID procesu je rovnaké ako UID vlastníka objektu alebo UID tvorcu objektu. To platí aj pre prístup skupiny DAC.

Prístup MAC je založený na SL procesu a objektu. Prístup MIC je založený na TL procesu a objektu.

Prístupové pravidlá pre obsahy objektov IPC sú rovnaké ako pre atribúty objektov IPC. Ak chcete čítať obsah alebo atribúty objektu IPC, vyžaduje sa prístup DAC READ, MIC READ a MAC READ. Na zápis do objektu IPC sa vyžaduje prístup DAC WRITE, MIC WRITE a MAC WRITE.

Atribúty objektu IPC sú tesnejšie obmedzené ako obsah objektu IPC. Zmena atribútov objektu IPC preto vyžaduje väčšie privilégiá. Na úpravu štandardných atribútov systému AIX, napríklad režimu, proces musí mať prístup DAC OWNER a MAC WRITE k objektu. Ak chcete zmeniť SL objektu IPC, proces musí mať všetky nasledujúce:

- Privilégium PV_SL_PROC
- DAC OWNER (len zúženie)
- DAC WRITE
- MAC WRITE
- Privilégium PV_SL_UG na rozšírenie SL alebo privilégium PV_SL_DG na zúženie SL
- PV_MAC_CL, ak existuje alebo nový SL je mimo vyúčtovania procesu
- MIC WRITE

Ak chcete zmeniť TL objektu IPC, proces musí mať všetky nasledujúce:

- Privilégium PV_TL
- DAC OWNER
- MAC WRITE
- MIC WRITE

Okrem toho, aby bolo možné zamknúť alebo odomknúť segment zdieľanej pamäte v pamäti, proces musí mať privilégium PV_KER_IPC_O. Proces tiež vyžaduje privilégium PV_KER_IPC na zmenu msg qbytes frontu správ v podprograme msgctl.

Súvisiace koncepty:

“Programovanie Dôveryhodný systém AIX” na strane 437
Systémová bezpečnosť závisí od softvéru, hardvéru a firmvéru dôveryhodnej výpočtovej bázy (TCB) a zahŕňa celé jadro operačného systému, všetky ovládače zariadenia a moduly System V STREAMS, rozšírenia jadra a všetky dôveryhodné programy. Všetky súbory používané týmito programami sa pri rozhodovaní o bezpečnosti tiež považujú za súčasť TCB.

Vytvorenie a vymazanie objektu IPC:

Neexistujú žiadne obmedzenia pre vytvorenie objektu IPC. Keď proces vytvorí objekt IPC, objekt zdedí SL a TL procesu.

Režim prístupu na objekt IPC musí zadať systémové volanie, pomocou ktorého sa objekt vytvorí.

Ak chcete vymazať objekt IPC, proces musí mať pre objekt oprávnenie DAC OWNER, MIC WRITE a MAC WRITE.

Práca v dôveryhodnej sieti:

Pre rozšírené atribúty bezpečnosti vylepšených bezpečnostných systémov je potrebný súbor požiadaviek na bezpečnú prácu v sieti. Dôveryhodná sieť AIX podporuje niekoľko uznávaných bezpečnostných štandardov pre prácu v sieti vrátane štandardu Ministerstva obrany USA RFC1108 RIPSO (Revised Internet Protocol Security Option) a CIPSO (Commercial Internet Protocol Security Option).

AIX zahŕňa podporu dôveryhodnej siete pre IPv4 aj IPv6. Pri komunikácii s ostatnými dôveryhodnými systémami je SL zapuzdrené vo voľbách IP podľa štandardov CIPSO/RIPSO. Kontroly MAC sa uplatňujú vo vrstve IP pre návestia SL, ktoré sa zasielajú alebo prijímajú v paketoch. Povolený rozsah návestí je nakonfigurovaný pomocou sieťových pravidiel. Sieťové pravidlá sa skladajú z pravidiel hostiteľa a pravidiel rozhrania. Dôveryhodná sieť systému AIX nainštaluje len predvolené pravidlá rozhrania (jedno pravidlo na nakonfigurované rozhranie). Ak chcete povoliť jemnejšie filtrovanie, môžete nakonfigurovať pravidlá hostiteľa. Ak chcete nakonfigurovať pravidlá hostiteľa aj rozhrania, môžete použiť príkaz netrule. Operácie podporované príkazom netrule zahŕňajú pridávanie, vymazávanie, vypisovanie a dotazovanie pravidiel.

Na inicializáciu podsystému dôveryhodnej siete a uchovávanie databázy pravidiel dôveryhodnej siete môžete použiť aj príkaz tninit.

Zakázanie konta Root:

Užívateľské konto Root je na systémoch Dôveryhodný systém AIX zakázané. Primárne je to kvôli minimalizácii škôd, ktoré môžu byť spôsobené systému jediným užívateľom so všetkými privilégiami.

Všetky typy prihlásení do systému ako užívateľ root sú zakázané. Prihlásenia užívateľa root umožňuje len príkaz **su**. Procesom vlastneným kontom root nie sú priradené žiadne špeciálne privilégiá. Programy typu setuid a non-setuid, vlastnené kontom root, pracujú ako predtým, keď boli spustené autorizovanými užívateľmi. Pre neautorizovaných užívateľov bude program bežať, ak režimové bity DAC alebo zoznamy ACL umožnia spustenie, ale program nebude mať priradené žiadne privilégiá, takže program nemusí byť schopný vykonať privilegované operácie, keď je spustený neautorizovanými užívateľmi. Preto je potrebné priradiť správne privilégiá novým nainštalovaným aplikáciám, od ktorých je očakávané vykonávanie privilegovaných operácií.

Úlohy administrácie systému môžu byť vykonávané užívateľmi, ktorým boli priradené roly Information System Security Officer (ISSO), System Administrator (SA), alebo System Officer (SO). Tieto roly umožnia ľubovoľnému užívateľovi vykonávať úlohy administrácie systému.

Poznámka: Počas inštalácie systému Dôveryhodný systém AIX je atribút **su** konta root nastavený na false. Ak chcete povoliť prístup na konto root iným administrátorom, musí užívateľ autorizovaný ako ISSO resetovať tento atribút na true použitím príkazu **chuser** a priradiť k tomuto kontu heslo.

Podpora návestí v auditovaní:

Primárnym účelom auditovacieho podsystému je monitorovanie a zaznamenávanie udalostí súvisiacich s bezpečnosťou.

Informácie, ktoré poskytol auditovací podsystém umožňujú zaznamenávanie nasledujúcich typov informácií:

- Pokus o narušenie bezpečnostnej politiky

- Úspešné dokončenie akcií súvisiacich s bezpečnosťou

Auditovací podsystém poskytuje nasledujúce schopnosti:
- Stanovenie udalostí, ktoré sa budú auditovať
- Zapnutie a vypnutie auditovania na spustenom systéme
- Neprerušované prepínanie (bez straty informácií) súborov preverovacích záznamov
- Konverzia informácií auditu do formy čitateľnej pre človeka
- Výber a spracovanie podmnožín informácií auditu

Keď nastavujete auditovací podsystém, ISSO by mal poznať, čo sa má auditovať, podmienky pri ktorých k auditovaniu dochádza a ako inicializovať a zastaviť auditovanie. Podrobné informácie o konfigurácii, spúšťaní a zastavovaní, administrácii a revízii auditu nájdete v téme Prehľad auditovania.

Auditovací podsystém si udržiava svoj aktuálny stav a je v tomto stave automaticky reštartovaný po poruche napájania, havárii systému, výpadku elektrického prúdu alebo po inom prerušení. Auditovací podsystém dokáže sám seba automaticky vypnúť, vypnúť systém alebo zmeniť súbory auditu, ak nastane stav, v ktorom už viac nedokáže ukladať záznamy auditu do existujúceho súboru auditu. Súbory auditu sa dajú automaticky prepnúť, keď sa súborový systém zaplní na požadovanú úroveň. Avšak v prípade katastrofálneho výpadku elektrického prúdu môže dôjsť k strate malého množstva záznamov auditu.

Viacúrovňové adresáre a adresáre s oddielmi:

Viacúrovňový adresár je štandardný adresár, ktorému je namiesto jedného označenia SL priradený rozsah označení SL. Adresár s oddielmi sa užívateľovi javí ako jediný adresár. Avšak súbory, ktoré sú užívateľovi zobrazené, sú v skutočnosti umiestnené v skrytých podadresároch adresára s oddielmi.

Viacúrovňové adresáre:

Viacúrovňový adresár je štandardný adresár, ktorému je namiesto jedného označenia SL priradený rozsah označení SL.

Aby mohol proces zobraziť názvy súborov vo viacúrovňovom adresári, musí pôsobiť na bezpečnostnej úrovni, ktorá je vyššia, než minimálne SL adresára. Aby mohol vytvoriť, alebo vymazať samotné súbory, musí proces pôsobiť v rámci rozsahu SL viacúrovňového adresára.

Každý súbor vo viacúrovňovom adresári má svoje vlastné SL a je chránený štandardnými obmedzeniami MAC. Každý proces, ktorý má do adresára prístup, však môže vidieť aj názvy objektov v tomto adresári. Preto môže mať proces schopnosti MAC čítať a zapisovať do adresára, ale nie je schopný čítať a/alebo zapisovať do niektorých súborov v adresári, a to aj napriek tomu, že názvy súborov v adresári zobraziť môže.

Adresáre s oddielmi:

Adresár s oddielmi sa užívateľovi javí ako jediný adresár. Užívateľovi zobrazené súbory sú však v skutočnosti umiestnené v skrytých podadresároch adresára s oddielmi.

Viacúrovňové adresáre predstavujú bezpečnostné riziko. Proces fungujúci na vysokej bezpečnostnej úrovni môže čítať súbor na nižšej bezpečnostnej úrovni, a potom vytvoriť súbory na rovnako vysokej bezpečnostnej úrovni. Zatiaľ čo funkcie MAC zamedzujú tomu, aby procesy s nižšou bezpečnosťou nové súbory čítali, stále môžu procesy s nižšou bezpečnosťou vidieť názvy týchto nových súborov. Ak proces s vysokou bezpečnosťou dal novým súborom názvy na základe obsahu pôvodného súboru s vysokou bezpečnosťou, mohol by proces s nižšou bezpečnosťou získať prístup k informáciám s vysokou bezpečnosťou tým, že prečíta názvy nových súborov.

Keď je vytvorený adresár s oddielmi a proces adresuje tento adresár, systém vytvorí skrytý podadresár s rovnakým SL, ako mal adresujúci proces. Ak tento proces vytvorí nejaký súbor, je tento súbor vlastne vytvorený v skrytom podadresári. Adresár s oddielmi môže obsahovať niekoľko takých skrytých podadresárov, ale proces pristupujúci

adresáru s oddielmi uvidí len súbory v skrytom podadresári s rovnakým SL, aké má pristupujúci proces. Keď proces vytvorí adresár potomka podadresára s oddielmi, adresár potomka je pod-podadresár s oddielmi.

Adresáru s oddielmi je priradený rozsah SL v rozmedzí od SYSTEM_LOW do SYSTEM_HIGH. Takže, k adresárom s oddielmi môže pristupovať každý proces.

Užívatelia s autorizáciou **aix.mls.pdir.mkdir** môžu adresáre s oddielmi vytvárať príkazom **pdmkdir**. Prázdny adresár s oddielmi je možné odstrániť príkazom **pdrmdir**. Pomocou príkazu **pdset** je možné zmeniť bežný adresár na typ adresára s oddielmi. Neexistuje príkaz, ktorým by bolo možné zmeniť adresár s oddielmi na bežný adresár.

V rámci adresára s oddielmi môžete vo všetkých existujúcich podadresároch s oddielmi, ktoré majú vyššie SL, vytvárať odkazy na súbor v inom podadresári toho istého adresára s oddielmi. To umožňuje prístup k súboru v rámci adresára s oddielmi všetkým procesom, ktoré majú prístup k tomu podadresáru s oddielmi, alebo k podadresárom vyššej úrovne v tom istom adresári s oddielmi. Na toto pripojenie súboru použiť príkaz **pdlink**.

Režimy prístupov k adresárom s oddielmi:

Každému procesu je pri jeho vytvorení priradený jeden z dvoch režimov, buď reálny režim, alebo virtuálny režim. Tento režim určuje, akým spôsobom bude proces zobrazovať adresáre s oddielmi.

Proces v reálnom režime narába s adresárom s oddielmi ako so štandardným viacúrovňovým adresárom. Ku všetkým podadresárom v oddieloch môže pristupovať ako k štandardným adresárom, na aké sa vzťahujú normálne obmedzenia DAC, MIC, a MAC. Proces v reálnom režime môže vstúpiť do adresára s oddielmi a zobraziť všetky podadresáre, na ktoré sa vzťahujú obmedzenia DAC, MIC a MAC.

Proces vo virtuálnom režime do adresára s oddielmi nikdy nevstupuje, ale je namiesto toho presmerovaný do podadresára v oddiele, ktorého maximálne aj minimálne SL sú rovnaké, ako aktuálne SL procesu.

Proces v reálnom režime môže pomocou príkazu **pdmode** spustiť príkaz vo virtuálnom režime (napríklad pdmode ls). Podobne aj proces vo virtuálnom režime môže, taktiež pomocou príkazu **pdmode**, spustiť príkaz v reálnom režime (napríklad pdmode -r ls). To však vyžaduje autorizáciu aix.mls.pdir.mode. S týmto oprávnením môžete spustením pdmode -r sh prepínať aj shell spustený vo virtuálnom režime na shell spustený v reálnom režime. Na spustenie programu vo virtuálnom režime počas reálneho režimu nie je potrebná žiadna autorizácia.

Zobrazovanie a zmena typov adresára:

Príkaz **lstxattr** sa používa na zobrazenie typu adresára ako súčasti atribútu secflags. FSF_PDIR označuje adresár s oddielmi, FSF_PSDIR označuje podadresár s oddielmi a FSF_PSSDIR označuje podadresár podadresára s oddielmi. Ak chcete zmeniť typ zvyčajného adresára na typ adresára s oddielmi, použite príkaz **pdset**.

Správa Dôveryhodný systém AIX

Riadenie systému Dôveryhodný systém AIX zahŕňa množstvo faktorov, ktoré sú pre Dôveryhodný systém AIX špecifické.

Inštalácia Dôveryhodný systém AIX

Dôveryhodný systém AIX môže byť povolený len počas inštalácie operačného systému pomocou voľby Security Model z inštalačnej ponuky.

Voľba migrácie pre Dôveryhodný systém AIX nie je podporovaná. Pri inštalácii typu Preservation súborový systém musí byť JFS2. Pri tichej sieťovej inštalácii si pozrite Tabuľka 37 na strane 408, kde nájdete heslá priradené predvoleným administrátorom.

Tabuľka 37. Heslá pre predvolených administrátorov

Užívateľ	Heslo
isso	isso
sa	sa
so	so

Prevádzkové režimy

Na umožnenie konfigurácie a údržby systému a na jeho každodennú prevádzku sú k dispozícii dva prevádzkové režimy - režim konfigurácie a operačný režim.

Keď systém nabootuje, na začiatku beží v režime konfigurácie. Po dokončení inicializácie je prevádzkový režim zmenený na operačný.

Režim konfigurácie sa používa na údržbu a obnovu systému. Keď je systém zavedený v režime jediného užívateľa, je systém minimálne nakonfigurovaný a je vypnutá sieť. Režim konfigurácie sa používa na administráciu kritických častí systému, ktoré súvisia s bezpečnosťou.

Operačný režim je štandardným režimom prevádzky systému. Systém sa prepne do tohto režimu po dokončení všetkých úloh, vyžadovaných na vstup do štandardnej úrovne prevádzky.

Prevádzkový režim systému môže byť zobrazený pomocou príkazu **getrunmode** a môže byť modifikovaný príkazom **setrunmode**.

Bezpečnostné príznaky jadra

Bezpečnostné príznaky jadra sa používajú na povolenie/zakázanie určitých bezpečnostných funkcií, ako napríklad vynútenie kontroly návestí, kontrola návestí integrity počas operácií načítavania, a na iné účely.

Jadro kontroluje bezpečnostné príznaky jadra predtým ako si vynúti bezpečnostné kontroly. Tieto príznaky sú podporované len vtedy, keď je povolený Dôveryhodný systém AIX. V užívateľskom priestore sú tieto príznaky uložené v databáze ODM. V závislosti od režimu spustenia systému, jadro skontroluje prítomnosť príslušných bezpečnostných príznakov jadra.

Tabuľka 38. Bezpečnostné príznaky jadra a predvolené hodnoty

Bezpečnostný príznak jadra	Povolený	Zakázaný	Predvolené nastavenie prevádzkového režimu	Predvolené nastavenie konfiguračného režimu
tnet_enabled	Funkčnosť dôveryhodnej siete je k dispozícii	Funkčnosť dôveryhodnej siete nie je možné nakonfigurovať alebo používať	Zakázaný	Zakázaný
tl_write_enforced	MIC vynútený pri operáciách zápisu, vymazania a premenovania	Konfigurácia je nastavená tak, že TL sa nemôžu používať pre kontroly zápisu	Povolený	Povolený
tl_read_enforced	MIC vynútený pri operáciách načítavania	Konfigurácia je nastavená tak, že TL sa nepoužívajú pre kontroly načítavania	Zakázaný	Zakázaný
sl_enforced	MAC vynútený	Konfigurácia je nastavená tak, že SL sa nepoužívajú pre kontroly prístupu	Povolený	Zakázaný
trustedlib_enabled	Príznak FSF_TLIB je akceptovaný v objektoch súborového systému	Príznaky FSF_TLIB nie sú akceptované	Zakázaný	Zakázaný

Nastavenie parametrov jadra

Jadro Dôveryhodný systém AIX môže byť nakonfigurované na presadzovanie bezpečnostných obmedzení podľa požiadaviek politík lokality.

Konfigurácia bezpečnosti môže byť zobrazená pomocou príkazu **getsecconf** a môže byť modifikovaná pomocou príkazu **setsecconf**. Konfigurovateľné sú tieto parametre jadra:

- Presadzovanie návestí citlivosti
- Presadzovanie čítania integrity
- Presadzovanie zápisu integrity
- Dôveryhodná sieť
- Dôveryhodná knižnica

Tieto parametre môžu byť nakonfigurované len kým je systém v režime konfigurácie.

Prispôsobovanie súboru /etc/security/enc/LabelEncodings

Označenia sú pre systém zadefinované v súbore /etc/security/enc/LabelEncodings a môžu byť prispôsobené pre každú lokalitu.

Po inštalácii Dôveryhodný systém AIX je možné prispôsobovať označenia.

Systém Dôveryhodný systém AIX má zadefinované nízke SL systému SYSTEM LOW SL (SLSL), ktoré je nižšie, než všetky ostatné označenia citlivosti v systéme a má zadefinované vysoké SL systému, SYSTEM HIGH SL (SHSL) ktoré je vyššie, než všetky ostatné označenia citlivosti. Podobne aj nízke TL označenie SYSTEM LOW TL (SLTL) je nižšie, než všetky ostatné označenia integrity v systéme a vysoké TL označenie SYSTEM HIGH TL (SHTL), ktoré je vyššie, než všetky ostatné označenia integrity. Tieto definície preberajú hodnoty vysokých a nízkych SL a TL, ako sú zadefinovaný v súbore /etc/security/enc/LabelEncodings.

Pri zavedení systému Dôveryhodný systém AIX sú označenia systému zo súboru /etc/security/enc/LabelEncodings zavedené do jadra. Tieto označenia je do jadra možné stiahnuť aj príkazom **setsyslab**. Označenia systému, tak ako sú zadefinované v jadre, je možné vypísať príkazom **getsyslab**. Po úprave súboru /etc/security/enc/LabelEncodings je odporúčané systém znova zaviesť.

Na miesta, kde sa začínajú kľúčové slová, je do súboru /etc/security/enc/LabelEncodings možné vkladať poznámky. Poznámky začínajú znakom * a pokračujú po koniec riadka.

Súbor /etc/security/enc/LabelEncodings obsahuje informácie o verzii a nasledujúce povinné časti. Každá časť by mala začínať jedným z týchto kľúčových slov nasledovaným dvojbodkou (:).

- classifications,
- information labels,
- sensitivity labels,
- clearances,
- channels,
- printer banners,
- accreditation range.

Súbor /etc/security/enc/LabelEncodings začína položkou VERSION. Táto položka je postupnosťou znakov a môže obsahovať medzery.

V časti môže byť uvedené každé z nasledujúcich kľúčových slov. Tieto kľúčové slová sú zakončené bodkočiarkou (;):

name=*name*
: Kľúčové slovo, ktoré definuje úplný názov klasifikácie alebo oddelenia.

sname=*name*
: Kľúčové slovo, ktoré definuje skrátený názov. Voliteľné.

aname=*name*
: Alternatívne kľúčové slovo pre klasifikáciu. Voliteľné.

value=*value*
Kľúčové slovo, ktoré definuje internú celočíselnú hodnotu klasifikácie alebo oddelenia.

compartments=*bit*
Kľúčové slovo, ktoré určuje, ktorý bit oddelenia musí byť 0, alebo 1, ak sa v označení nachádza slovo.

Vylepšenia Dôveryhodný systém AIX vo formáte kódovania označení

Kódovanie označovaní, tak ako je predpísané nariadením Defense Intelligence Agency Document DDS-2600-6216-93, nepodporuje označenia integrity.

Označenia citlivosti sú štandardne používané ako označenia integrity. Dôveryhodný systém AIX poskytuje podporu voliteľnej časti s označeniami integrity, ktorá môže byť odlišná, než časti s označeniami citlivosti. To poskytuje flexibilitu v možnosti používať rozličné klasifikačné názvy a hodnoty pre označenia citlivosti a integrity. Označenia citlivosti môžu mať napríklad predponu SL a označenia integrity predponu TL, ako v nasledujúcich tabuľkách:

Tabuľka 39. Názvy a hodnoty klasifikácií označení citlivosti

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

Tabuľka 40. Názvy a hodnoty klasifikácií označení integrity

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

Na časť s označeniami integrity sa vzťahujú nasledovné pravidlá:

- Časť "INTEGRITY LABELS" by mala byť pridaná iba za časťou "NAME INFORMATION LABELS". V prípade, že administrátor nezadefinoval voliteľnú časť "NAME INFORMATION LABELS" by mala byť časť "INTEGRITY LABELS" pridaná za časťou "ACCREDITATION RANGE".
- V súbore s kódovaním označení by mala byť len jedna časť "INTEGRITY LABELS". Na objekty a subjekty sa vzťahuje tá istá časť.
- Nová časť "INTEGRITY LABELS" je voliteľná časť. V prípade, ak táto časť neexistuje, mala by byť použitá klasifikácia tak, ako je daná v povinnej časti "CLASSIFICATIONS".
- Časť "INTEGRITY LABELS" by mala byť podobná, ako časť "CLASSIFICATIONS". Obsahovala by nasledujúce kľúčové slová: **"name="**, **"sname="**, **"aname="**, a **"value="**. Kľúčové slová **"initial compartments="** a **"initial markings="**, ktoré sú súčasťou časti "CLASSIFICATIONS" nebudú platné s časťou "INTEGRITY LABELS".
- Rozsah údajov pre **"value="** by bol rovnaký, ako ten v časti "CLASSIFICATIONS" – od minima 0 po maximum 32000.

Spustenie systému

Systémová bezpečnosť sa automaticky vyvolá počas sekvencie spustenia systému. Skontrolujte, či sú parametre bezpečnosti pre daný systém zobrazené počas sekvencie spúšťania správne.

Konfiguračný spúšťací režim:

Konfiguračný režim je využívaný pri údržbe a obnove systému.

Keď je systém zavedený v režime jediného užívateľa, je systém minimálne nakonfigurovaný a je vypnutá sieť.

Prevádzkový spúšťací režim:

Prevádzkový režim je využívaný pri každodenných operáciách.

Normálne by mal systém zaviesť priamo do viacužívateľského režimu. Ak program autorizácie pri zavedení dostane platné meno užívateľa a heslo, vstúpi systém do prevádzkového režimu, je zobrazená obrazovka autentifikácie prihlásenia do konzoly a platní užívatelia sa môžu prihlásiť.

Bezpečnostné mechanizmy, akými sú označenia citlivosti, DAC (discretionary access control), MAC (mandatory access control), kontroly privilégií, identifikácia a autentifikácia, ako aj autorizácie, sú aktívne v konfiguračnom režime, aj prevádzkovom režime, tak ako to prikazujú príslušné príznaky bezpečnostnej konfigurácie. Bližšie informácie nájdete v príkaze **getsecconf**.

Odporúčaným riešením je pracovať so systémom len v prevádzkovom režime, čím zabezpečíte dostupnosť celkovej očakávanej funkčnosti systému.

Proces zavedenia:

Do súboru /etc/inittab v systémoch Dôveryhodný systém AIX boli pridané nové zavádzacie skripty. Tieto nové zavádzacie skripty sú rc.mls.boot, rc.mls.net a rc.mls a sú vykonávané v tomto poradí.

V skripte rc.mls.boot sú vykonané tieto kroky:

1. Je spustená interaktívna kontrola integrity, ktorá vyzve užívateľa, aby poskytol informácie, ako spracovať každý rozpor (pomocou príkazu **trustchk**).
2. Sú nastavené príznaky zabezpečenia konfiguračného režimu jadra (pomocou príkazu **setsecconf**).
3. Sú nastavené označenia (minimálne a maximálne označenia citlivosti a označenia integrity).
4. Príznaky zabezpečenia konfiguračného režimu jadra sú zobrazené na obrazovke.

V skripte rc.mls.net sú vykonané tieto kroky:

1. Inicializujte podsystém Dôveryhodný systém AIX.
2. Ak existuje súbor /etc/security/rules.int, zavedie do jadra databázu pravidiel.

V skripte rc.mls sú vykonané tieto kroky:

1. Inicializujte podsystém Dôveryhodný systém AIX.
2. Ak existuje súbor /etc/security/rules.int, zavedie do jadra databázu pravidiel.

Poznámka: Akákoľvek zmena týchto zavádzacích sriptov môže spôsobiť zlyhanie systému.

Prispôsobovanie spúšťania systému:

Aj napriek tomu, že to nie je odporúčané, je možné vypnúť autentifikáciu pri zavedení a kontrolu integrity systému pri spúšťaní systému.

Pokiaľ nie je vypnutá autentifikácia pri zavedení a kontrola integrity systému, musí byť pri spustení systému operátor fyzicky prítomný pri systémovej konzole.

Vypnutie autentifikácie BOOT:

Autentifikáciu BOOT môžete vypnúť zadaním príkazu **rmitab bootauth** alebo pomocou ponuky SMIT.

Vypnutie kontrol integrity systému:

Automatickú kontrolu integrity pri zavedení systému môžete vypnúť tým, že zo skriptu **rc.mls.boot** odstránite riadok trustchk.

Vypínanie systému

Vypnutie systému je privilegovaná operácia a je chránená autorizáciou aix.system.boot.shutdown.

Systém môže vypnúť ľubovoľný užívateľ s rolou SO alebo inou rolou, ktorá má túto autorizáciu.

Dôveryhodná obnova

Môže sa vyskytnúť situácia, že sa systém vypne v nečistom stave. K tomuto môže dôjsť v dôsledku výpadku prúdu, náhodného vypnutia systému alebo zlyhania hardvéru. Dôveryhodný systém AIX môže byť z týchto situácií obnovený bez mimoriadnych procedúr opätovného zavedenia.

Keď sa systém reštartuje, všetky ochranné mechanizmy budú aktívne bez ohľadu na to, akým spôsobom bol systém vypnutý. Počas procedúry spúšťania systému sa ešte pred prihlásením užívateľov automaticky skontrolujú všetky súborové systémy kvôli poškodeniu. Skripty spúšťania spustia príkaz **fsck**, ktorý zabezpečí alebo zneprístupní pre neoprávnených užívateľov poškodené alebo ohrozené súbory.

Príkaz **trustchk** zaznamená všetky nezrovnalosti v bezpečnostných atribútoch súborov alebo adresárov a interaktívne vyzve užívateľa na opravu týchto atribútov. Príkaz **trustchk** spúšťajte pri každej možnosti ohrozenia integrity súborového systému. Bližšie informácie získate pomocou príkazu **trustchk**.

Prihlásenie

Každý užívateľ Dôveryhodný systém AIX by mal mať priradené náležité previerky citlivosti a integrity, aby sa mohol prihlásiť do systému.

Užívateľove previerky sú v súbore /etc/security/user definované ako užívateľské atribúty. Atribúty minsl a maxsl definujú previerku citlivosti užívateľa. Atribúty mintl a maxtl definujú previerku integrity užívateľa. Atribúty defsl a deftl definujú efektívne úrovne citlivosti a integrity užívateľa pri prihlásení.

Atribúty previerok užívateľa sa dajú modifikovať pomocou príkazov chuser a chsec a ich zoznam sa dá vypísať pomocou príkazov lsuser a lssec.

Užívatelia si môžu vypísať zoznam svojich vlastných návestí, ale nemôžu ich zmeniť. Ak chcete vypísať zoznam úrovní previerok iných užívateľov, užívateľ musí mať autorizáciu aix.mls.clear.read. Ak chcete previerky modifikovať, užívateľ musí mať autorizáciu aix.mls.clear.write.

Ak sa chcete prihlásiť, všetky nasledujúce pravidlá nadradenosti musia byť platné:

- Hodnota defsl musí byť nadradená hodnote minsl
- Hodnota maxsl musí byť nadradená hodnote defsl
- Hodnota deftl musí byť nadradená hodnote mintl
- Hodnota maxtl musí byť nadradená hodnote deftl

Požadované efektívne úrovne citlivosti a integrity môžete zadať počas prihlásenia pomocou volieb **–e** a **–t** príkazu **login**. Bližšie informácie nájdete v príkaze **login**.

Ak sa chcete prihlásiť na úroveň citlivosti, ktorá nie je v akreditačnom rozsahu systému, musíte mať autorizáciu aix.mls.label.outsideaccred.

Dôveryhodný systém AIX neumožňuje prihlásenie systémových užívateľov (užívatelia s kratším uid ako 128).

Príčiny zlyhaní prihlasovania

Pokus o prihlásenie môže zlyhať z mnohých dôvodov.

Pokus o prihlásenie zlyhá, ak platí čokoľvek z nasledujúcich:

- Bolo zadané neplatné prihlasovacie ID
- Bolo zadané neplatné prihlasovacie heslo
- Konto je označené ako uzamknuté, lebo počet predchádzajúcich nesprávnych pokusov o prihlásenie pre toto konto prekračuje limity systému
- Prihlasovací port je označený ako uzamknutý, lebo počet predchádzajúcich nesprávnych pokusov o prihlásenie pre tento port prekračuje limity systému
- Prihlasovacie ID nemá platné vyrovnanie
- Zadané návestie (alebo predvolené návestie citlivosti alebo integrity pre prihlasovacie ID, ak nebolo zadané žiadne návestie) nie je platné, nie je v rámci vyrovnania pre prihlasovacie ID, nie je v rámci vyrovnania pre prihlasovacie zariadenie, alebo nie je v rámci rozsahu akreditácie systému
- Užívateľ nemá prístup DAC na názov cesty prihlasovacieho programu prostredia Shell, alebo konto užívateľa nemá prístup DAC na spustenie na prihlasovací program prostredia Shell
- Užívateľ nemá prístup MAC alebo MIC na názov cesty prihlasovacieho programu prostredia Shell, alebo konto užívateľa nemá prístup MAC alebo MIC na čítanie na prihlasovací program prostredia Shell
- UID prihlasovacieho ID bol menší ako 128

Prepnutie užívateľa pomocou príkazu su

Ak je v systéme Dôveryhodný systém AIX vyvolaný príkaz **su** s voľbou **-**, úroveň vymazania súčasného užívateľa musí byť vyššia ako úroveň vymazania nového užívateľa.

Pre návestia citlivosti a integrity musia byť splnené tieto podmienky:

- maximálne vymazanie súčasného užívateľa musí byť vyššie než maximálne vymazanie nového užívateľa
- minimálne vymazanie nového užívateľa musí byť vyššie než minimálne vymazanie súčasného užívateľa
- platné vymazanie súčasného užívateľa musí byť nižšie než maximálne vymazanie nového užívateľa a maximálne vymazanie nového užívateľa musí byť vyššie než minimálne vymazanie nového užívateľa.

Povinnosti užívateľskej bezpečnosti

Užívateľ si musí byť vedomý určitých povinností a pochopiť a dodržiavať ich. Užívatelia musia uchovávať svoje heslá v tajnosti, hlásiť zmeny stavu, podozrenia z porušenia bezpečnosti atď.

Heslá

Heslá je potrebné zapamätať si a nezapisovať ich na žiadne médium. Ak heslo získa iný užívateľ, môže to ohroziť bezpečnosť informácií v systéme.

Najbežnejšou hrozbou pre bezpečnosť hesiel je ich odhalenie. Najjednoduchším spôsobom ochrany konta pred útokom neoprávneného užívateľa, ktorý mohol odhaliť heslo, je pravidelná zmena hesla. Častou zmenou hesiel znížite pravdepodobnosť odhalenia počas životnosti jednotlivých hesiel. Čím dlhšie sa heslo používa, tým viac možností existuje na jeho odhalenie.

Ak majú užívatelia možnosť vybrať si svoje vlastné heslá, nové heslo musí mať dĺžku minimálne šiestich znakov a musí obsahovať aspoň dva abecedné a jeden numerický znak. Heslo by nemalo odrážať žiadnu osobnú ani profesionálnu stránku užívateľa (napríklad priateľov, meno užívateľa, meno domáceho zvieraťa alebo pracovný titul) a

nemalo by ísť o bežné slovo, ktoré sa dá vyhľadať v slovníku. Schémy slúžiace na uhádnutie hesla často prehľadajú jeden alebo viacero slovníkov a dôležitý zoznam osobných položiek, napríklad meno užívateľa, mená detí alebo domácich zvierat a narodeniny.

Heslá môžu mať obmedzenú životnosť stanovenú užívateľom s oprávnením ISSO. Ak platnosť hesla uplynula a užívateľ sa pokúsi prihlásiť, dostane oznam, že musí zmeniť heslo a môže sa prihlásiť, pokiaľ heslo nie je zmenené. Odporúča sa meniť heslá užívateľov častejšie, než je uvedená životnosť hesla. Ak vznikne akékoľvek podozrenie z ohrozenia užívateľovho hesla, je potrebné zmeniť ho ihneď.

Ponechanie systému bez dozoru

Ak je nejaký užívateľ prihlásený do aktívnej relácie, nikdy nenechávajte systém bez dozoru. Ak musíte odísť od počítača, hoci aj nakrátko, odporúča sa, aby ste sa pred odchodom odhlásili zo systému.

Riadenie zabezpečeného systému

Riadenie zabezpečeného počítačového systému zahŕňa vytvorenie a presadenie politík bezpečnosti a pravidelné monitorovanie systému.

Nasledujúci zoznam by mal slúžiť ako počiatočný bod pre vývoj politiky riadenia zabezpečených prostriedkov pre vašu lokalitu:

- Maximálna úroveň zabezpečenia v rozsahu akreditácie systému by nemala byť vyššia ako maximálna úroveň zabezpečenia lokality, v ktorej je systém umiestnený.
- Systémový hardvér by mal byť na zabezpečenom mieste. Najbezpečnejšie umiestnenia sú všeobecne miestnosti v interiéri, ktoré nie sú na prízemí.
- Fyzický prístup k systémovému hardvéru by mal byť obmedzený, monitorovaný a dokumentovaný.
- Zálohy systému a archivačné médiá by mali byť uložené na bezpečnom mieste, oddelenom od lokality systémového hardvéru. Fyzický prístup na toto miesto by mal byť obmedzený podobným spôsobom, ako prístup k systémovému hardvéru.
- Prístup k prevádzkovým manuálom a administračnej dokumentácii by mal byť obmedzený pre tých, ktorí to potrebujú vedieť.
- Rebooty systému, zlyhania napájania a vypnutia by mali byť zaznamenané. Poškodenie súborového systému by malo byť dokumentované a všetky postihnuté súbory by mali byť analyzované kvôli možným narušeniam bezpečnostnej politiky.
- Inštalácia nových programov, či už importovaných alebo vytvorených, by mala byť obmedzená a monitorovaná. Nové programy by mali byť pred ich spustením pozorne preskúmané a otestované.
- Neobvyklé alebo neočakávané správanie akéhokoľvek systémového softvéru by malo byť dokumentované a oznámené a mala by byť zistená príčina daného správania.
- Ak je to možné, mali by systém spravovať minimálne dvaja ľudia. Jedna osoba by mala mať rolu isso a druhá by mala mať rolu sa.
- Privilégium PV_ROOT by nemalo byť použité. Na administráciu systému by malo byť dostačujúce spúšťanie privilegovaných programov užívateľmi ISSO, SA alebo SO .
- Auditovacie informácie by mali byť zhromažďované do protokolov a mali by byť pravidelne vyhodnocované. Nepravidelné alebo neobvyklé udalosti by mali byť poznamenané a mala by byť vyšetrená ich príčina.
- Mal by byť minimalizovaný počet prihlásení s rolami isso, sa a so.
- Mal by byť minimalizovaný počet programov setuid a setgid a mali by byť používané len v chránených podsystémoch.
- Privilégiá priradené novým programom by mali byť stanovené a minimalizované vyhodnotením tých, ktoré sú priradené už existujúcim programom.
- Bezpečnostné atribúty súborov a adresárov by mali byť pravidelne overované pomocou príkazu **trustchk**.
- Všetky heslá by mali obsahovať najmenej 8 znakov. Toto by mal pravidelne overovať užívateľ ISSO.
- Všetci užívatelia by mali mať platné prihlasovacie prostredie Shell. Toto by mal pravidelne overovať užívateľ SA.

- ID bežných užívateľov by nemali byť systémovými ID. Toto by mal pravidelne overovať užívateľ SA. Systémové ID je také, ktoré má UID nižšie ako 128.

Konfigurácia systému:

Na správnu konfiguráciu systému musia užívatelia s oprávnením ISSO a SA vykonať určité kroky. Užívateľ s oprávnením ISSO primárne zodpovedá za riadenie bezpečnosti a užívateľ s oprávnením SA za každodennú správu.

Užívateľ s oprávnením ISSO vykonáva tieto úlohy:

- Nainštaluje a nakonfiguruje základné funkcie bezpečnosti vrátane systémového auditu, účtovníctva a bezpečnosti pre alokovateľné zariadenia.
- Upravuje skripty spúšťania systému v súboroch /etc/rc.mls a /etc/rc.mls.boot na splnenie bezpečnostnej politiky danej lokality.

  Poznámka: Niektoré zmeny vykonané v skriptoch spúšťania systému nie sú súčasťou otestovanej konfigurácie a je potrebné otestovať ich ešte pred akreditáciou systému.
- Nakonfiguruje celosystémové prihlasovacie parametre.
- Nakonfiguruje celosystémové parametre hesiel.
- Nakonfiguruje rozsah návestí SL pre zariadenia tty, ktoré umožnia užívateľom prihlásiť sa do rozsahov SL zadaných pre port tty. Bližšie informácie získate pomocou príkazu **chsec**.
- Nakonfiguruje návestia SL systémového zariadenia pre páskové a disketové jednotky. Bližšie informácie získate pomocou príkazu **setsecattr**.
- Nakonfiguruje systémové funkcie bezpečnosti nakonfigurovateľné na lokalite.

  Poznámka: Zmeny vykonané v nakonfigurovateľných funkciách bezpečnosti nie sú súčasťou otestovanej konfigurácie a je potrebné otestovať ich ešte pred akreditáciou systému. Zmena predvolených nastavení konfigurácie môže viesť k menej bezpečnému režimu systémovej prevádzky.
- Nakonfiguruje dôveryhodnú bezpečnostnú databázu pre dôveryhodné zavedenie a obnovu. Bližšie informácie získate pomocou príkazu **trustchk**.
- Nakonfiguruje v systéme užívateľské skupiny.

Užívatelia s oprávnením ISSO a SA spolupracujú pri konfigurácii tlačiarní. Užívateľ s oprávnením SA nakonfiguruje tlačiarne pre systém a užívateľ s oprávnením ISSO nakonfiguruje rozsah návestí SL pre tlačiarne.

Sieťová konfigurácia:

ISSO je primárne zodpovedné za zabezpečenie siete a SA je primárne zodpovedné za každodennú administráciu siete. ISSO a SA spájajú svoje sily, aby bola sieť správne nakonfigurovaná.

Počas inštalácie Dôveryhodný systém AIX je zabezpečenie siete konfigurované s predvolenými nastaveniami. Môže tiež posunúť označenia citlivosti ďalším hostiteľom Dôveryhodný systém AIX v sieti. ISSO inštaluje a konfiguruje základnú funkcionalitu siete, ktorú systém poskytuje. ISSO konfiguruje sieťové tabuľky a spúšťaním príkazu **tninit** ukladá databázy.

Sieťový prístup:

Ak sa k inému, než Dôveryhodný systém AIX systému pripájate prostredníctvom siete, alebo k systému Dôveryhodný systém AIX, ktorý nepoužíva funkciu Trusted Networking, niektoré bezpečnostné atribúty možno nebudú odvysielané do systému, ktorý nie je Dôveryhodný systém AIX. V takom prípade systém Dôveryhodný systém AIX použije predvolené bezpečnostné mechanizmy. Predvolené bezpečnostné mechanizmy vytvára administrátor systému.

Konfigurácia užívateľského konta:

Užívatelia s oprávnením ISSO a SA pri konfigurácii užívateľských kont v systéme spolupracujú. ISSO primárne zodpovedá za riadenie bezpečnostných užívateľských atribútov a SA za ostatné užívateľské atribúty.

ISSO vykonáva pre každého užívateľa tieto úlohy:
- Nakonfiguruje vymazanie. Bližšie informácie získate pomocou príkazov **chsec** a **chuser**
- Nakonfiguruje roly a oprávnenia
- Nakonfiguruje užívateľské skupiny
- Nastaví úroveň vymazávania domovského adresára. Bližšie informácie získate pomocou príkazu **settxattr**.
- Nastaví heslo
- Nastaví masku auditu

SA vykonáva tieto úlohy:
- Nakonfiguruje užívateľské kontá
- Informuje ISSO o nových užívateľských kontách, ktoré vyžadujú bezpečnostné atribúty

Konfigurácia súborového systému:

V Dôveryhodný systém AIX je podporovaná väčšina systémov, avšak podpora pre rozšírené atribúty súvisiace s bezpečnosťou Dôveryhodný systém AIX na objektoch súborového systému je k dispozícii len na JFS2 s EAv2.

Súborový systém JFS2 s EAv1 sa skonvertuje do EAv2, keď sa pripája k systému Dôveryhodný systém AIX. Súbory na týchto súborových systémoch JFS2 nemajú bezpečnostné atribúty. Systém na prístup do týchto súborov používa predvolené atribúty SYSTEM_LOW. Bezpečnostné atribúty môžete na súboroch nastaviť pomocou príkazu **settxattr**.

V sieťovom prostredí môže byť adresár na jednom systéme označený ako zdieľaný, čo znamená, že adresár môže byť pripojený k a môže sa naň pristupovať z iných systémov v sieti ako by to bol koreňový adresár súborového systému na oddiele lokálneho disku.

Súborový systém môže byť viacúrovňový súborový systém (MLFS) alebo jednoúrovňový súborový systém (SLFS). V MLFS má každý objekt súboru svoje vlastné návestia, zatiaľ čo všetky objekty v SLFS majú rovnaké návestia ako bod pripojenia. SLFS nepodporuje viacúrovňové adresáre a adresáre s oddielmi.

Prístup k súborovému systému:

Keď sa proces pokúsi pristúpiť na objekt súborového systému, systém si overí prístup ku každému komponentu názvu cesty.

Ak proces nemá oprávnenie na vyhľadávanie vo všetkých adresároch v názve cesty, tento proces nebude môcť na objekt pristúpiť. Ak sa použije relatívna cesta, skontroluje sa prístup k aktuálnemu adresáru bez ohľadu na to, či sa explicitne odkazuje na aktuálny adresár znakom bodky (.) na začiatku cesty.

Riadenie dôveryhodnej siete:

Existuje množstvo aspektov riadenia dôveryhodnej siete, ktoré zahŕňajú konfiguráciu a konfiguračnú databázu, syntax príkazu netrule a špecifikáciu pravidiel, príznaky dôveryhodnej siete a voľby RIPSO/CIPSO.

Varovanie kvôli predvolenej konfigurácii:

Sieťové možnosti systému AIX Trusted Network boli starostlivo navrhnuté, aby umožnili akúkoľvek mysliteľnú želanú konfiguráciu. Zmena predvolených hodnôt konfigurácie bez pochopenia AIX Trusted Network však môže byť nebezpečná.

Je možné, že pri nesprávnom nakonfigurovaní počítača, automaticky znížite, rozšírite, alebo úplne odstránite bezpečnostné informácie. Preto by ste predvolené hodnoty v sieťových tabuľkách nemali meniť, kým nie ste s produktom AIX Trusted Network dôverne oboznámený.

Konfiguračná databáza systému AIX Trusted Network:

Konfigurácia siete v čase zavedenia sa vytvára pomocou súborov rules.host a rules.int.

Po predvolenej inštalácii Dôveryhodný systém AIX neexistujú žiadne pravidlá pre hostiteľov ani súbory pravidiel. Príkazom **netrule** s použitím príznaku **-u** je možné ukladať nové alebo aktualizované pravidlá do súborov. Tieto súbory sú binárne databázy, ktoré sa ovládajú príkazom **tninit**. Užívateľ musí mať autorizáciu aix.mls.network.init, aby mohol používať príkaz **tninit**.

Zobrazenie databázy pravidiel produktu AIX Trusted Network:

Obsah databázy pravidiel produktu AIX Trusted Network je možné zobraziť akciou **disp** príkazu **tninit**.

Rozšírenia **.host** a **.int** zadaním nasledujúceho príkazu doplníte o *filename*, aby ste mohli vygenerovať názvy súborov databázy pravidiel hostiteľa a rozhranie databázy pravidiel. Obsah oboch súborovo bude v čitateľnej forme odoslaný štandardnému toku výstupu.

```
tninit disp filename
```

Zadaním nasledujúceho príkazu zobrazíte predvolenú konfiguráciu zavedenia:

```
tninit disp /etc/security/rules
```

Zavedenie databázy pravidiel AIX Trusted Network:

Príkaz **tninit** načíta sadu databáz pravidiel AIX Trusted Network a zavedie ich do jadra, aby sa stali aktívnou sadou. Názvy súborov hostiteľa a tabuľky akreditácie rozhrania sú zadané v rovnakej metóde ako akcia **tninit disp**.

Voliteľný príznak **-m** zadáva, že systém by mal zachovať existujúce pravidlá hostiteľov. Ak príznak **-m** nie je zadaný, všetky existujúce pravidlá hostiteľov sa odstránia ešte pred zavedením novej aktívnej sady. Ak je príznak **-m** zadaný, nastane agregácia existujúcich a nových sád pravidiel hostiteľov, pričom ak dôjde ku konfliktu, nové pravidlá prepíšu existujúce pravidlá. Všetky pravidlá rozhrania sa prepíšu, či bude alebo nebude príznak **-m** zadaný.

Nasledujúci príkaz zavedie nové pravidlá, a zároveň zachová sady starých pravidiel:

```
tninit -m load /dir/dir/filename
```

Tento príkaz použil súbor, zadaný cez parameter *filename* a pripojí k nemu prípony **.host** a **.int**, aby vytvoril dva súbory, z ktorých databáza pozostáva.

Ukladanie databázy pravidiel dôveryhodnej siete AIX:

Na zavedenie a uloženie databázy pravidiel sa používa podobná sémantika.

Ľubovoľný zadaný názov je doplnený príponou **.int** a **.host** na vytvorenie dvoch súborov, používaných na uloženie databázy. Akcia ukladania príkazu **tninit** uloží všetky pravidlá, ktoré sú aktuálne aktívne v jadre.

Ak chcete vytvoriť štandardnú skupinu pravidiel, musíte použiť príkaz **netrule** na prispôsobenie pravidiel jadra požadovanej politike pravidiel lokality a potom spustiť príkaz **tninit**. Nasledujúci príkaz vytvorí súbory /etc/security/rules.int a /etc/security/rules.host:

```
tninit save /etc/security/rules
```

Konfigurácia jadra AIX Trusted Network:

Pomocou príkazu **netrule** môžete kompletne nakonfigurovať množinu pravidiel jadra AIX Trusted Network tak, aby vyhovovala bezpečnostnej politike lokality, ak máte autorizáciu aix.mls.network.config.

Príkaz **netrule** je možné používať na manipuláciu s pravidlami rozhrania hostiteľa aj siete v jadre. Bližšie informácie nájdete v príkaze **netrule**.

Každé rozhranie v systéme musí mať priradené pravidlo. Ak sa pokúsite vymazať pravidlo rozhrania, vráti sa do svojho predvoleného stavu. Ak pridáte ďalšie pravidlo rozhrania, nové pravidlo rozhrania prepíše aktuálne pravidlo. Predvolené pravidlo rozhrania je možné zobraziť dotazovaním pravidla rozhrania s názvom rozhrania “default.” Napríklad: # netrule iq default

Syntax príkazu netrule:

Pravidlá syntaxe príkazu **netrule** sa týkajú hostiteľov a rozhraní.

Ak je príkaz **netrule** použitý pre hostiteľa, má nasledujúcu syntax:

netrule h l [**i** | **o** | **io**]

netrule h q { **i** | **o** } *src_host_rule_specification dst_host_rule_rule_specification*

netrule h - [{ **i** | **o** } [**u**] [*src_host_rule_specification dst_host_rule_specification*]

netrule h + { **i** | **o** } [**u**] *src_host_rule_specification dst_host_rule_specification* [*flags*] [*RIPSO/CIPSO_options*] *security*

Ak je príkaz **netrule** použitý pre rozhranie, má nasledujúcu syntax:

netrule i l

netrule i q *interface*

netrule i + [**u**] *interface* [*flags*] [*RIPSO/CIPSO_options*] *security*

Prvý element (h, alebo i) naznačuje, či ide o operáciu hostiteľa, alebo sieťového rozhrania.

Požadovaná akcia vymenovaná nižšie. Dostupné sú štyri rozdielne akcie:

l Vymenuje všetky pravidlá.

q Dotaz na konkrétne pravidlo.

- Odstráni pravidlo hostiteľa, alebo vráti pravidlo rozhrania do jeho pôvodného stavu.

+ Pridá, alebo prepíše pravidlo.

Tretí element v pravidlách hostiteľa určuje typ pravidla. Pri pravidlách hostiteľa existuje rozdiel medzi prichádzajúcimi a odchádzajúcimi pravidlami. Pravidlá na vstupe sú použité na všetky prichádzajúce pakety, zatiaľ čo pravidlá na výstupe sú použité na všetky odchádzajúce pakety; i naznačuje pravidlo na vstupe, o naznačuje pravidlo na výstupe a ak je to použiteľné, io alebo žiaden príznak naznačujú pravidlo pre vstup aj výstup. Ak je pri pridávaní, alebo odstraňovaní pravidla hostiteľa, alebo rozhrania zadaný posledný element u, sú po úspešnom pridaní alebo odstránení pravidla hostiteľa, alebo pravidla rozhrania, aktualizované súbory /etc/security/rules.host a /etc/security/rules.int.

Špecifikácia pravidiel AIX Trusted Network:

Pravidlá rozhrania vyžadujú zadať názov sieťového rozhrania. Pravidlá hostiteľa sú flexibilnejšie a preto vyžadujú zložitejšiu špecifikáciu pravidiel.

Ak chcete zadať rozhranie, zadajte názov sieťového rozhrania, na ktoré sa má aplikovať pravidlo. Názvy sieťových rozhraní sú názvy ako en0. Názvy sieťových rozhraní zobrazíte pomocou príkazu **ifconfig -a**. Konkrétne rozhranie musíte zadať len podľa názvu. Nemôžete zadať port, protokol alebo masku podsiete.

Pravidlá hostiteľa vyžadujú zložitejšie špecifikácie pravidiel. Systém AIX Trusted Network používa najpodrobnejšie použiteľné pravidlo. Napríklad, politika stránky sa dá nakonfigurovať tak, aby pravidlo hostiteľa s maskou 24 platilo pre všetkých hostiteľov na podsieti, ale na jedného hostiteľa na sieti sa môže aplikovať podrobnejšie pravidlo a tento hostiteľ bude používať podrobnejšie pravidlo. Ďalšie podrobnejšie pravidlo sa tiež dá aplikovať na jeden konkrétny port TCP na tomto hostiteľovi. Flexibilita konfigurácie systému AIX Trusted Network vám poskytuje schopnosť implementovať čokoľvek, podľa potrieb bezpečnostnej politiky vašej lokality pre aplikáciu. Presná syntax:

zdrojový_hostiteľ [*/maska*] [**=** *protokol*] [**:***zač_číslo_portu* [**:***konc_číslo_portu*]]

cieľový_hostiteľ [*/maska*] [**=** *protokol*] [**:***zač_číslo_portu* [**:***konc_číslo_portu*]]

zdrojový_hostiteľ
: Názov hostiteľa, adresa IPv4 alebo adresa IPv6 zdrojového hostiteľa.

cieľový_hostiteľ
: Názov hostiteľa, adresa IPv4 alebo adresa IPv6 cieľového hostiteľa.

maska
: Maska podsiete. Číslo označuje, koľko bitov z MSB je platných. Keď sa zapíše dvojica adresy IPv4/podsiete, *a*.*b*.*c*.*d*/*e*, *e* je číslo od 0 do 32. Toto číslo určuje počet jednotiek na začiatku masky podsiete. Napríklad, pre adresu IPv4 hodnota /24 špecifikuje masku podsiete 255.255.255.0, ktorá, pri 32-bitovom zobrazení, je 11111111.11111111.11111111.00000000. To je 24 jednotiek a za nimi osem núl.

protokol
: Číslo alebo názov protokolu podľa záznamu v súbore /etc/protocols (napríklad =tcp).

zač_číslo_portu
: Buď port TCP alebo UDP, na ktorý sa pravidlo aplikuje alebo začiatok rozsahu, ak sa pravidlo aplikuje na rozsah portov. To môže byť buď číslo portu alebo názov služby UDP alebo TCP podľa záznamu v súbore /etc/services.

konc_číslo_portu
: Horná hranica rozsahu portov.

Popis príznaku systému AIX Trusted Network:

Systém AIX Trusted Network má dva klastre príznakov. Ak tieto nie sú špecifikované, použijú sa predvolené hodnoty.

Príznaky **-d** a **-r** sa používajú nasledujúcim spôsobom:

-d *drop*

drop Systém AIX Trusted Network je možné nakonfigurovať tak, aby zrušil všetky pakety

- **r** Zrušiť všetky pakety na tomto rozhraní
- **n** Automaticky nerušiť všetky pakety na tomto rozhraní (predvolená voľba rozhrania)
- **i** Použiť predvolenú voľbu rozhrania (predvolený hostiteľ, len hostiteľ)

-f*rflag***:***tflag*

rflag Požiadavka na bezpečnostnú voľbu na prichádzajúcich (prijímaných) paketoch

r Len RIPSO

c Len CIPSO

e Buď CIPSO alebo RIPSO

n Ani CIPSO ani RIPSO (predvolené pre systém)

a Bez obmedzení

i Použiť predvolené pre rozhranie/systém (predvolené)

tflag Spracovanie bezpečnostnej voľby na odchádzajúcich (odosielaných) paketoch

r RIPSO sa aplikuje na hlavičky IP všetkých odchádzajúcich paketov

c CIPSO sa aplikuje na hlavičky IP všetkých odchádzajúcich paketov

i Použiť predvolenú voľbu rozhrania (predvolený hostiteľ, len hostiteľ)

Možnosti RIPSO/CIPSO:

Podsystém dôveryhodnej siete AIX podporuje voľby na konfiguráciu označovania paketov typu CIPSO a RIPSO.

-rpafs=*PAF_field* [**,** *PAF_field ...*]

Určuje každé pole *PAF_field*, ktoré je akceptované, keď sú prijaté pakety IPSO. Môže byť prítomných až 256 týchto polí.

-epaf=*PAF_field*

Určuje pole *PAF_field*, ktoré je pripojené na chybové odozvy, keď sú chybové pakety odosielané použitím IPSO na prenášaných paketoch.

-tpaf=*PAF_field*

Určuje pole *PAF_field*, ktoré je použité na odchádzajúce pakety, keď je na vysielané pakety použitý IPSO.

*PAF_field***:NONE |** *PAF* [**+** *PAF ...*]

Pole *PAF_field* je kolekciou niekoľkých *PAF*. Existuje päť samostatných *PAF*, ktoré môžu byť zahrnuté v jedinom poli *PAF_field*. Sú to tieto: **GENSER**, **SIOP-ESI**, **SCI**, **NSA** a **DOE**. Pole *PAF_field* je kombináciou týchto hodnôt, oddelených znamienkom plus (**+**). Napríklad pole *PAF_field*, ktoré obsahuje **GENSER** a **SCI**, je reprezentované ako **GENSER+SCI**. Môže byť použité špeciálne pole *PAF_field* **NONE**; to určuje *PAF_field* bez akýchkoľvek nastavených *PAF*.

-DOI=*doi*

Určuje doménu interpretácie pre pakety CIPSO. Prichádzajúce pakety CIPSO musia mať tento **DOI** a odchádzajúce pakety CIPSO budú týmto **DOI** označené.

-tags=*tag*[**,***tag ...*]

*tag***=1 | 2 | 5**

Určuje sadu značiek, ktoré sú akceptované a dostupné na prenos pomocou volieb CIPSO. Toto je kombináciou hodnôt **1**, **2** a **5**, oddelených čiarkami. Napríklad **1,2** by povolil značky **1** a **2**.

Bezpečnostná politika systému AIX Trusted Network:

Je potrebné zadať minimum povolených SL, maximum povolených SL a predvolený počet SL.

Implicitný alebo predvolený počet SL sa aplikuje na všetky pakety a nenesie žiadne informácie o ich vlastnom počte SL. Úrovne sa zadávajú pomocou nasledujúcej syntaxe:

+*min* **+***max* **+***predvolené*

Je možné použiť všetky návestia, ktoré sú platné podľa súboru kódovaní návestí. Pre návestia obsahujúce medzery nie je potrebné použiť úvodzovky.

Príklady príkazu netrule:

Toto sú príklady príkazu **netrule**.

Ak zadáte nasledujúci príkaz, nakonfigurujete **en0** aby nevypol žiadne možnosti bezpečnosti a aby cez neho prechádzali všetky pakety:

```
netrule i+ en0 +impl_lo +ts all +impl_lo
```

Nasledujúcim spôsobom nakonfigurujete hostiteľa **185.0.0.62**, aby akceptoval len pakety CIPSO v rozsahu od **CONFIDENTIAL A** po **TOP SECRET ALL**:

```
netrule h+i 192.168.0.0 /24 185.0.0.62 -fc:c +confidential a +top secret all +confidential a
```

Zadaním nasledujúceho príkazu pustíte všetky telnetové pakety z podsiete:

```
netrule h+i 192.168.0.0 /24 =tcp :telnet 192.0.0.5 -dr +impl_lo +impl_lo +impl_lo
```

Bližšie informácie o príkladoch nájdete v príkaze **netrule**.

Riadenie užívateľských kont:

Identifikačné a autentifikačné (I&A) informácie o každom užívateľovi sú chránené a používajú sa na jednoznačnú identifikáciu užívateľa a kontrolu užívateľových prístupových oprávnení v systéme.

Informácie o identite užívateľa zahŕňajú meno užívateľa, názov textu prihlasovacieho ID, ID užívateľa, ID skupiny, domovský adresár, heslo, parametre starnutia hesla, prostredie shell, vymazávania, oprávnenia a masku auditu. Väčšina užívateľských informácií je uložená v týchto súboroch:

/etc/passwd
: Mená užívateľov, ID užívateľov, pridelenia primárnej skupiny a domovské adresáre

/etc/group
: Pridelenia sekundárnej skupiny a domovské adresáre

/etc/security/passwd
: Užívateľské heslá v zašifrovanom tvare

/etc/security/user
: Prihlasovacie obmedzenia, parametre hesla (napríklad minimálna dĺžka), umaska atď.

Súbory /etc/security/passwd a /etc/security/user nie sú pre bežných užívateľov čitateľné. Súbor /etc/security/passwd je chránený bez zapnutých ľubovoľných prístupových bitov a SL SYSTEM_HIGH. Ochrana pred prečítaním zašifrovaného hesla bežnými užívateľmi eliminuje následné šifrovacie/porovnávacie rutiny, ktoré sa pokúšajú o porovnávanie zašifrovaného hesla.

Oprávnení užívatelia môžu upravovať tieto súbory priamo, ale je často výhodnejšie použiť na úpravu užívateľských parametrov príkaz **smit**. Príkaz **smit** vyvolá nástroj SMIT (System Management Interface Tool), ktorý zobrazí ponuky s výbermi pre úlohy systémového riadenia, napríklad údržbu užívateľa.

ID užívateľa a skupiny:

Existujú dve triedy užívateľských ID: systémové ID a ID bežného užívateľa. Systémové ID sú rezervované pre vlastníctvo chránených podsystémov a špeciálne funkcie systémovej správy. ID bežného užívateľa sa prideľujú jednotlivcom, ktorí používajú systém interaktívne.

Každý užívateľ má jedinečný užívateľský ID, ktorý sa používa na jeho identifikáciu v systéme. Každému užívateľovi môže byť pridelený aj jeden alebo viacero skupinových ID. Skupinové ID sú zdieľané užívateľmi v rovnakej skupine a nemusia byť nutne jedinečné. Existujú limity rozsahov numerických hodnôt používaných pre ID. Nasledujúca tabuľka definuje limity rozsahu ID. Hodnoty boli definované tak, aby povoľovali dostatočný počet systémových ID, ID bežného užívateľa a skupinových ID.

ID systémového užívateľa
0 až 127

ID bežného užívateľa
128 až MAXUID

ID bežnej skupiny
0 až MAXUID-1

Hodnota MAXUID je definovaná v súbore /usr/include/sys/param.h

Pri prideľovaní hodnôt užívateľských ID novým užívateľom treba postupovať opatrne. Ak je bežnému užívateľovi neúmyselne pridelená hodnota užívateľského ID menšia než 128, užívateľ sa nebude môcť prihlásiť do systému.

Hodnoty užívateľských ID by sa nemali používať opakovane. Keď je užívateľ vymazaný, odporúča sa, aby položky zostali v súboroch /etc/passwd a /etc/security/passwd a aby bolo konto zamknuté. Môžete to vykonať pomocou príkazu **smit**. Predídete tak prihláseniu užívateľa a opakovanému použitiu ID. Ochranou pred opakovaným použitím ID predchádzate tomu, aby nový užívateľ vstupoval do súborov, ktoré patria predchádzajúcemu užívateľovi a ktoré možno ešte neboli odstránené. Umožňuje to tiež jednoznačne zrekonštruovať stopu auditu.

Súbory /etc/passwd, /etc/security/passwd a /etc/group môžete riadiť pomocou príkazov **mkuser**, **chuser**, **rmuser**, **pwdadm** a **passwd**. Tieto príkazy uplatňujú všetky vyššie uvedené opatrenia, ako aj všetky ostatné aspekty systémovej bezpečnosti. Príkaz **mkuser** môže pridávať do systému len bežných užívateľov.

Poznámka: Dôsledne uplatňujte tieto zásady:
- Nikdy neprideľujte znova predtým použitý užívateľský ID novému užívateľovi
- Nikdy neprideľujte duplicitné užívateľské ID
- Nikdy neprideľujte systémový ID bežnému užívateľovi
- Nikdy neprideľujte MAXUID ako užívateľský alebo skupinový ID

Heslá:

Heslo je textový reťazec, ktorý je priradený užívateľovi a pomocou ktorého je užívateľ autentifikovaný pri spustení relácie.

Heslo je uložené v zašifrovanej forme v tieňovom súbore. V systéme nie sú uložené žiadne heslá, ktoré by neboli šifrované.

Poznámka: Heslá užívateľov s rolami sú z hľadiska bezpečnosti systému mimoriadne dôležité a mali by byť vždy chránené.

Starnutie hesla:

Užívatelia môžu meniť svoje heslá, pokiaľ sú dodržané kritériá starnutia hesiel.

Starnutie hesla vyžaduje, aby užívatelia zmenili svoje heslá, ak ich heslo v systéme existovalo počas zadefinovaného časového úseku. Starnutie hesla zahŕňa časový úsek minimálneho a maximálneho veku hesla. Heslo nemôže byť zmenené predtým, než uplynie časový úsek minimálneho veku hesla. To isté heslo musí byť zmenené, po uplynutí časového úseku jeho maximálneho veku.

Parametre starnutia hesla môžu byť nastavené v súbore /etc/security/user. So starnutím hesla súvisia nasledujúce parametre:

maxage
Maximálny počet týždňov, počas ktorých je heslo platné.

maxexpired
Maximálny počet týždňov po hodnote maxage, počas ktorých môže byť heslo užívateľom zmenené.

minage
Minimálny počet týždňov, medzi zmenami hesiel.

minlen Minimálna dĺžka hesla

Je možné nastaviť ďalšie parametre určujúce znaky, ktoré sú v hesle povolené. Úplný zoznam parametrov hesiel nájdete v príkaze **passwd**.

Shell:

Počas práce v aplikácii, akou je textový editor alebo tabuľka, užívatelia všeobecne nepotrebujú priamo pracovať s operačným systémom, keďže túto interakciu riadi aplikácia. Avšak niektorí užívatelia potrebujú priamu interakciu s operačným systémom bez rozhrania inej aplikácie.

Keď je potrebná priama interakcia s operačným systémom, užívatelia musia použiť program prostredia Shell. Program prostredia Shell umožňuje užívateľom zadávať príkazy AIX a priamo pristupovať na súbory a adresáre a vykonávať ďalšie operácie. Každý užívateľ musí mať zadaný predvolený program prostredia Shell vo svojom súbore /etc/passwd. Užívateľov predvolený program prostredia Shell (napríklad /bin/sh, /bin/csh alebo /bin/ksh) je spúšťaný príkazom **login** alebo **xterm**, keď užívateľ potrebuje použiť prostredie Shell.

Prihlasovacie efektívne SL a TL:

Užívatelia majú priradené predvolené prihlasovacie SL a TL. Predvolené prihlasovacie SL a TL sú efektívne SL a efektívne TL užívateľovho procesu po úspešnom prihlásení.

Ak sa užívateľ nechce prihlásiť svojim predvoleným SL, môže si pomocou voľby **-e** príkazu **login** v čase prihlasovania vybrať iné SL. Užívateľom zadané SL musí byť nadradené užívateľovej previerke a musí byť z rozsahu akreditácie systému. TL môže užívateľ zadať v čase prihlasovania s použitím voľby **-t** príkazu **login**.

Predvolené prihlasovacie SL a TL sú definované v súbore /etc/security/user spoločne s menom užívateľa a previerkou každého užívateľa. Efektívne SL užívateľa musí byť zo SL rozsahu tty, ktorý je zadaný v súbore /etc/security/login.cfg. Maximálne SL tty musí byť nadradené nad užívateľovým efektívnym SL a to musí byť nadradené nad minimálnym SL. Efektívne TL užívateľa musí byť rovnaké ako TL pre tty.

Platný rozsah:

Počas prihlásenia je prostrediu shell procesu užívateľa priradených šesť označení.

Efektívne SL využíva systém pri kontrolách MAC. Minimálny platný rozsah SL a maximálny platný rozsah SL obmedzuje efektívne SL; efektívne SL nemôže byť vyššie, než maximálny platný rozsah SL a musí byť vyššie, než minimálne SL. Efektívne TL využíva systém pri kontrolách MIC. Minimálny platný rozsah TL a maximálny platný rozsah TL obmedzuje efektívne TL; efektívne TL nemôže byť vyššie, než maximálny platný rozsah TL a musí byť vyššie, než minimálne TL.

Užívateľ s oprávnením ISSO môže upravovať platný rozsah SL, platný rozsah TL, predvolené prihlasovacie SL a predvolené prihlasovacie TL každého užívateľa. Tieto hodnoty sú zadefinované v súbore /etc/security/user.

Rozdelenie zodpovednosti za užívateľské informácie:

Jediný užívateľ nemôže do systému pridať iného užívateľa. Užívatelia sú do systému pridávaní kombinovanou akciou užívateľov s autorizáciou SA a ISSO.

Užívateľ s autorizáciou SA môže pridať užívateľské informácie nesúvisiace s bezpečnosťou, ktoré zahŕňajú názov užívateľa, ID užívateľa, ID skupiny, názov textu prihlasovacieho ID, prostredie shell a domovský adresár. Užívateľ s autorizáciou ISSO môže pridať užívateľské informácie súvisiace s bezpečnosťou, ktoré zahŕňajú heslo užívateľa, platný rozsah, masku auditu a roly. Požiadavka, aby užívateľa museli pridať dvaja ľudia, zabraňuje tomu, aby jediný užívateľ s autorizáciou pridelil inému užívateľovi autorizáciu do celého systému.

Vylepšené auditovanie:

Dôveryhodný systém AIX má vylepšený auditovací podsystém tak, aby mohol získavať dodatočné bezpečnostné podrobnosti.

Nové polia v zázname auditu:

Do všetkých záznamov auditu systému AIX pre Dôveryhodný systém AIX boli pridané nasledujúce polia. Tieto nové polia môžu byť v príkaze **auditselect** použité ako kritériá výberu.

- roly auditovaného procesu,
- efektívne TL auditovaného procesu, alebo objektu,
- efektívne SL auditovaného procesu, alebo objektu,
- efektívne privilégiá auditovaného procesu.

V niektorých protokoloch auditu Dôveryhodný systém AIX vykonáva aj audit nasledujúcich bezpečnostných atribútov:

- TL auditovaného procesu, alebo objektu,
- SL auditovaného procesu, alebo objektu,
- Príznaky bezpečnosti súvisiace so systémom Dôveryhodný systém AIX.

Tieto nové bezpečnostné atribúty môžete zobraziť príkazom **auditpr -v**.

Rozsahy auditov:

Dôveryhodný systém AIX obsahuje mechanizmus, ktorý administrátorovi umožňuje v závislosti na TL a/alebo SL auditovaných procesov alebo objektov stanoviť skupinu auditovacích rozsahov. Všetky objekty a subjekty, ktorých TL alebo SL sa nachádzajú mimo auditovacieho rozsahu, budú ignorované.

Auditovací rozsah procesov a objektov nastavíte pridaním stanzy **war** do súboru /etc/security/audit/config:

```
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
```

Premenné **obj_min_sl** a **obj_max_sl** určujú rozsah auditovaných SL objektov. Premenné **sub_min_sl** a **sub_max_sl** určujú rozsah auditovaných SL subjektov (procesov). **obj_min_tl** a **obj_max_tl** určujú auditovaný rozsah TL objektov. **sub_min_tl** a **sub_max_tl** určujú auditovaný rozsah TL subjektov (procesov).

Stanzu **war** číta príkaz **audit start** a zavedie ju do jadra ešte predtým, než je spustený auditovací podsystém. Ak je stanza **war** vynechaná, budú aktuálne rozsahy auditovania z jadra odstránené. Ak nie je v jadre určený žiaden TL SL rozsah auditovania, nevykonáva jadro žiadne kontroly TL alebo SL rozsahov auditovania.

Príznak jadra Dôveryhodný systém AIX:

Keď je systém v momente inštalácie nakonfigurovaný ako systém Dôveryhodný systém AIX, v premennej **_system_configuration** sa aktivuje globálny príznak jadra. V jadre je k dispozícii makro **__MLS_KERNEL()** na určovanie, či je systém nakonfigurovaný ako systém Dôveryhodný systém AIX. Toto makro môžu volať aplikácie v užívateľskom priestore alebo rutiny jadra. Návratová hodnota **1** z makra **__MLS_KERNEL()** označuje, že systém je nakonfigurovaný ako systém Dôveryhodný systém AIX. Všetky ostatné návratové hodnoty označujú, že systém nie je nakonfigurovaný ako systém Dôveryhodný systém AIX.

Aktualizácia existujúcich programov:

Existujúce privilegované alebo dôveryhodné programy vo všeobecnosti fungujú správne na dôveryhodnom systéme bezo zmeny.

Je však možné vykonať určité zmeny na zlepšenie úrovne dôvery a/alebo kompatibility smerom nahor pre tieto programy. Mnohé z odporúčaní pre vytváranie nových programov platia aj pre aktualizáciu existujúcich programov. Zvlášť platia nasledujúce odporúčania:

- Programy, ktoré testovaním zisťujú, či sú privilegovanými procesmi (t.j., či efektívne ID užívateľa je 0), je potrebné upraviť v súlade s pokynmi v dokumente Priama kontrola privilégií
- Kód, ktorý manipuluje so štandardnými bitmi oprávnení systému UNIX (bity režimu), je potrebné zmeniť tak, aby odrážali možnú existenciu ACL
- Kód, ktorý sa použil na spustenie ako setuid-to-root, je potrebné skontrolovať, či používa privilégiá a musí mať priradené príslušné privilégiá

Zálohovanie a obnova:

Import a export údajov v systémoch Dôveryhodný systém AIX vyžíva dôveryhodné verzie príkazov **backup** a **restore**.

Príkazy **backup** a **restore** boli rozšírené na spracovanie označení. Tieto rozšírenia sú pre užívateľa transparentné a bez ohľadu na rozšírenia označovania tieto príkazy fungujú rovnako, ako štandardné príkazy AIX **backup** a **restore**. Ak chcete v rozšírených bezpečnostných informáciách vypnúť backup alebo restore, môžete použiť príznak **-O**.

Systém importovania/exportovania je chránený kombináciou mechanizmov privilégií a autorizácie.

Obmedzenia príkazu cron:

Ak je systém v konfiguračnom režime, je príkaz **cron** zakázaný a nebude spúšťať žiadne úlohy. Ak je systém v prevádzkovom režime, spúšťa príkaz **cron** úlohy s takým označením citlivosti, v akej bol príkaz vydaný a s predvoleným označením integrity užívateľa.

Existujú obmedzenia, ako napríklad minimálny a maximálny platný rozsah užívateľa. V závislosti na tom, čo je aktuálnejšie, je platný rozsah prevzatý buď z nastavení v čase, keď bol príkaz vydaný, alebo z času, keď bol príkaz **cron** ostatný raz reštartovaný. Príkaz **cron** môže administrovať len užívateľ SA.

Pripojenie a prístup k súborovým systémom:

Dôveryhodný systém AIX podporuje označovanie (SL a TL) na súborových systémoch JFS2 s EAv2. Ak je to nevyhnutné, môže SA, alebo SO pripojiť súborový systém, ktorý toto označovanie nepodporuje (CDFS or HSFS). V takom prípade žiaden zo súborov pripojeného súborového systému nemá vlastné SL, TL, ani FSF, ale namiesto toho zdedí bezpečnostné atribúty bodu pripojenia.

Riadenie systému Dôveryhodný systém AIX

Ak chcete zabezpečiť bezpečnosť systému, dodržiavajte návod na správne riadenie systému Dôveryhodný systém AIX.

Riadenie systému Dôveryhodný systém AIX vykonávajú niektorí užívatelia, ktorých kontá majú priradené administračné roly. Títo užívatelia sa nazývajú ISSO (Information System Security Officer), SA (System Administrator) a SO (System Officer) a každý z týchto užívateľov má autorizácie, ktoré im umožňujú vykonávať špecifickú podmnožinu administračných úloh. Títo užívatelia majú priradené systémom definované roly isso, sa a so. Výrazy ISSO, SA a SO sa používanú na označenie užívateľov, ktorí majú roly isso, sa a so. Niektoré administračné povinnosti môžu vykonávať len dvaja z troch manažérov systému, ktorí pracujú spoločne, pretože jeden manažér, vystupujúci samostatne, nemá dostatočné oprávnenia na vykonávanie týchto povinností. Napríklad, keď do systému pridávate nového užívateľa, len SA môže pridať nové konto užívateľa a len ISSO môže vytvoriť užívateľove heslo, previerku a masku auditu. Toto rozdelenie práce poznáme ako pravidlo dvoch ľudí.

Poznámka: Efektívnosť pravidla dvoch ľudí závisí od autorizácií, ktoré majú priradené administračné roly. Priradenie viacerých autorizácií administračným roliam ako je potrebné môže oslabiť systém voči útokom z vnútra. Bližšie informácie o priraďovaní autorizácií roliam nájdete v RBAC.

Systémom definované roly isso, sa a so sú štandardne priradené nasledujúcim Dôveryhodný systém AIX autorizáciám. Zmenám týchto priradení venujte náležitú starostlivosť, pretože takéto zmeny môžu systém oslabiť.

Tabuľka 41. Roly a autorizácie

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		

Riadenie systému pre ISSO (Information System Security Officers):

Systém Dôveryhodný systém AIX je riadený koordinovanými aktivitami užívateľov ISSO, SA a SO.

Počas inštalácie Dôveryhodný systém AIX sa vytvoria tri predvolené kontá užívateľov **isso**, **sa** a **so** (ak tieto kontá ešte neexistujú, v prípade migrácie z bežného AIX do Dôveryhodný systém AIX). Títo užívatelia majú priradené isso, sa a so.

Poznámka: Predvolené kontá sú určené len pre úvodné nastavenie a konfiguráciu systému Dôveryhodný systém AIX. Odporúča sa, aby ste tieto roly priradili iným bežným užívateľom. Keď budú tieto roly priradené iným užívateľom, predvolené konto užívateľa môžete odstrániť. Bližšie informácie o inštalácii Dôveryhodný systém AIX nájdete v téme *Installation and migration*.

ISSO aktivity

Hlavnou zodpovednosťou ISSO (Information System Security Officer) je správa bezpečnosti systému. Len užívateľ s autorizáciou ISSO môže vykonávať aktivity ISSO. K týmto aktivitám patrí:

- Plánovanie, implementácia a presadzovanie bezpečnostnej politiky lokality
- Vytváranie celosystémových štandardných nastavení pre previerky užívateľov, autorizácie, privilégiá, prihlasovacie ovládacie prvky a parametre hesiel
- Nastavenie profilov autentifikácie užívateľov odráža úroveň dôveryhodnosti vkladanú do užívateľov, keď administrátor systému vytvára kontá užívateľov
- Priradenie bezpečnostných atribútov, SL a TL zariadeniam, ako napríklad terminály, tlačiarne, vymeniteľné diskové jednotky a magnetické páskové jednotky
- Priradenie bezpečnostných príznakov, návestí, privilégií a sád autorizácií súborom
- Obnovenie systému do dôveryhodného stavu v prípade zlyhania systému

Správa systému auditu:

Prístup k príkazom auditovania je obmedzený na užívateľov s autorizáciou **AUDITSYS**. Bližšie informácie nájdete v príkazoch **audit**, **auditselect** a **auditpr**.

Nasledujúci príklad ukazuje:

1. Ako vytvoriť súborový systém, ktorý sa má používať pre súbory stôp auditu
2. Ako spustiť systém auditu
3. Ak aktivovať generovanie niektorých záznamov
4. Ako analyzovať stopu auditu na získanie rôznych typov záznamov.

Spustite nasledujúce príkazy ako užívateľ s autorizáciou **FSADMIN**:

```
/usr/sbin/crfs -v jfs -g rootvg -m /audit -a size=32M -A yes
mount /audit
```

Pomocou príkazu **/tbin/auctlmod -e** pridajte nasledujúcu položku do sekcie užívateľov súboru /etc/security/audit/config:

```
meno_užívateľa = ALL
```

Nahraďte *meno_užívateľa* menom skutočného užívateľa, ktorý sa môže prihlásiť do systému.

Ako užívateľ ISSO vytvorte súbor s názvom /tmp/top_secret a zmeňte SL tohto súboru na **TS ALL**.

```
touch /tmp/top_secret
/usr/sbin/settxattr -f sl= "TS ALL" /tmp/top_secret
```

Spustite nasledujúci príkaz ako užívateľ s autorizáciou **AUDITSYS**:

```
/usr/sbin/audit start
```

Systém auditu je teraz nastavený a spustený tak, aby zaznamenával akcie užívateľa zadaného cez *meno_užívateľa*, keď sa tento užívateľ prihlási do systému.

Prihláste sa do systému s užívateľom zadaným cez *meno_užívateľa* v súbore /etc/security/audit/config a spustite nasledujúce príkazy:

```
ls -l /tmp/top_secret
exit
```

Spustite nasledujúce príkazy ako užívateľ s autorizáciou **AUDITSYS**:

```
audit shutdown
```

```
$ /usr/sbin/auditselect -e "mac_fail==WILDCARD" /audit/trail | \
/usr/sbin/auditpr -v -APSV > /tmp/audit_trail-mac_failure
```

Skontrolujte stopu auditu, ktorá bola presmerovaná na súbor /tmp/audit_trail-mac_failure a vyhľadajte **mac_fai**l. Auditselect bol modifikovaný, aby akceptoval nasledujúce voľby:

- **subj_sl**
- **obj_sl**
- **mac_fail**
- **mac_pass**
- **mic_fail**
- **mic_pass**
- **priv_fail**
- **priv_pass**
- **auth_pass**
- **fsf_fail**
- **fsf_pass**

Všetky tieto voľby používajú slovo **WILDCARD** ako hodnotu zhody.

Riadenie návestí objektov a procesov:

Každý objekt súborového systému a systémový proces má priradené návestia.

Všetky objekty súborového systému s výnimkou bežných súborov majú rozsah návestí citlivosti a návestie integrity. Procesy majú aj rozsah návestí citlivosti aj rozsah návestí integrity. Procesy majú okrem rozsahov aj efektívne SL a efektívne TL. Toto návestie označuje aktuálne SL alebo TL, v ktorom je proces spustený. Návestia si môžete prezrieť pomocou príkazu **lstxattr**. Návestia objektov a procesov súborového systému môžete nastaviť pomocou príkazu **settxattr**.

Riadenie zabezpečenia siete:

AIX Trusted Network vyžaduje, aby ISSO zadefinoval niekoľko tabuliek. Tieto tabuľky sú uložené v adresári /etc/security. Pomocou príkazu **tninit** je vytvorená binárna verzia, ktorá je potom zavedená do jadra.

Pravidlá hostiteľa a sieťového rozhrania určujú, ako systém zaobchádza s prichádzajúcimi a odchádzajúcimi paketmi. Pravidlá hostiteľa sú použité na konkrétnych hostiteľov. Pravidlá sieťových rozhraní sú použité na rozhrania, cez ktoré sú títo hostitelia pripojení k sieti. Ak dôjde k akýmkoľvek konfliktom medzi pravidlom hostiteľa a pravidlom rozhrania, prednosť má pravidlo hostiteľa.

Pomocou príkazu **netrule** môžete pravidlá pridať, upraviť a dotazovať. Vo všeobecnosti sa pravidlá týkajú použitých protokolov, rozsahov adries (tak hostiteľov, ako rozhraní) na ktorých sú pravidlá použité a toho, akým SL majú byť označené pakety. Bližšie informácie nájdete v príkaze **netrule**.

Pomocou príkazu **tninit** môžete inicializovať podsystém AIX Trusted Network, aby ste pravidlá mohli uložiť v binárnom formáte a zobraziť v textovom formáte.

Konfigurovateľné funkcie bezpečnosti:

Konfigurovateľné funkcie bezpečnosti sa zobrazujú počas sekvencie zavedenia.

Konfigurovateľné nastavenia sú uložené v ODM. Tieto nastavenia môžu byť zobrazené pomocou príkazu **getsecconf** a môžu byť modifikované užívateľom ISSO pomocou príkazu **setsecconf**.

Riadenie návestí:

ISSO užívateľ môže pridať, modifikovať alebo vymazať kódovania návestí cez modifikáciu súboru /etc/security/enc/LabelEncodings. Súbor /etc/security/enc/LabelEncodings definuje ako sa budú pre človeka čitateľné názvy mapovať do binárneho zobrazenia návestí systémovej citlivosti.

Poznámka: Výsledkom modifikovania súboru kódovaní návestí citlivosti v spustenom systéme môžu byť neplatné návestia, pokiaľ nebude zabezpečená extrémna starostlivosť. Keďže objekty môžu mať návestie zložené z jedného slova alebo z presne vymedzenej kombinácie slov, ľahkomyseľná zmena, pridanie alebo vymazanie presne vymedzených slovných kombinácií môže mať za následok neplatné návestia.

Súbor /etc/security/enc/LabelEncodings bude preložený do binárnej formy pomocou rutiny knižnice **l_init** a uložený do tabuliek. Tieto tabuľky sa používajú na skonvertovanie SL, tlačových transparentov a previerok do a zo svojich interných binárnych kódovaní.

Dôveryhodný systém AIX používa ako základ pre implementáciu označovania návestiami softvér MITRE Compartmented Mode Workstation Labeling. Dokument Compartmented Mode Workstation Labeling: Encodings Format, DDS-2600-6216-93 (MTR 10649 revision 1), september 1993 vysvetľuje štandardný formát kódovaní návestí.

Štandardný formát kódovania návestí zaobchádza s návestiami integrity a s návestiami citlivosti rovnako ako sa uvádza v časti **Sensitivity Labels** súboru /etc/security/enc/LabelEncodings.

Dôveryhodný systém AIX voliteľne podporuje časť **Integrity Labels**, ktorá umožňuje odlíšenie návestí integrity od návestí citlivosti.

Riadenie adresárov s oddielmi:

Normálnemu užívateľskému procesu sa adresár s oddielmi javí a funguje rovnako, ako bežný adresár. Pri adresári s oddielmi však rozličné procesy s rozličnými SL vidia v tom istom adresári odlišný obsah.

Ak napríklad proces, ktorý je spustený s bezpečnostnou nálepkou **SECRET** vytvorí v adresári s oddielmi súbor nazvaný **foo**, potom druhý proces spustený s bezpečnostnou nálepkou TOP SECRET tento súbor **foo** v adresári nevidí, ani k nemu nemôže pristupovať. Zároveň môže druhý proces vytvoriť svoj vlastný súbor **foo** bez toho, aby mu prekážal prvý súbor **foo**.

Toto je možné vďaka skrytým podadresárom. Pre každé jedinečné SL, s ktorým proces pristupuje k adresáru s oddielmi, existuje podadresár s oddielmi. Keď proces pristupuje k adresáru s oddielmi, presmeruje ho systém automaticky do skrytého podadresára. Vo vyššie uvedenom príklade sú vlastne dva súbory **foo** v rozličných podadresároch, aj napriek tomu, že sa užívateľovi javia akoby v tom istom adresári.

Bližšie informácie o adresároch s oddielmi nájdete v “Adresáre s oddielmi” na strane 406.

Adresáre s oddielmi sú podporované v JFS2 s EAv2.

Vytvorenie adresára s oddielmi:

Predvolený rozsah SL pri vytvorení adresára s oddielmi je nízke SL systému a vysoké SL systému. Pri prístupe k adresáru s oddielmi jadro automaticky vytvorí adresár potomka špecifický pre dané označenie (ak ešte žiaden neexistuje) a do tohto adresára potomka presmeruje užívateľský proces.

Adresár s oddielmi vytvorte pomocou príkazu **pdmkdir**. Aby mohol príkaz **pdmkdir** prepísať obmedzenia DAC, MAC a MIC, vyžaduje autorizáciu **aix.mls.pdir.create**. Pomocou príkazu **pdrmdir** môžete odstrániť prázdny adresár s oddielmi.

Podadresáre a pod-podadresáre s oddielmi

Adresáre potomkov adresára s oddielmi, vytvorené podľa konkrétnych označení, sú podadresáre s oddielmi. Keď proces vytvorí adresár potomka pod podadresárom s oddielmi (pomocou príkazu **mkdir**), tento adresár potomka je pod-podadresárom s oddielmi.

Keď je vytvorený podadresár s oddielmi, zdedí bezpečnostné atribúty svojho rodičovského adresára s oddielmi, okrem minimálneho SL a maximálneho SL. Minimálne a maximálne SL sú nastavené na efektívne SL procesu virtuálneho režimu, ktorý k tomuto podadresáru s oddielmi pristupoval ako prvý.

Dôveryhodný systém AIX pozná štyri rozdielne typy adresárov:
- bežný adresár (dir),
- adresár s oddielmi (pdir),
- podadresár s oddielmi (psdir),
- pod-podadresár s oddielmi (pssdir).

Virtuálny a reálny režim:

Existujú dva rôzne režimy prístupu do adresára s oddielmi: virtuálny a reálny.

Vo virtuálnom režime proces vstupujúci do adresára s oddielmi uvidí len obsah podadresára s oddielmi špecifický pre dané návestie. Adresár s oddielmi nie je nikdy viditeľný pre proces spustený vo virtuálnom režime. Adresár s oddielmi je viditeľný pre proces spustený v reálnom režime. Procesy spustené v reálnom režime uvidia celý reálny obsah adresárov s oddielmi aj podadresárov s oddielmi. Pre procesy spustené v reálnom režime systém nevykoná žiadne presmerovanie.

Procesy sú štandardne spustené vo virtuálnom režime. Reálny režim je určený len na účely správy súborového systému. Príkaz **pdmode** sa používa na spúšťanie príkazov v režime, ktorý je iný než režim prostredia shell aktuálneho procesu alebo na prepnutie do prostredia shell v inom režime.

Aj keď proces užívateľa v reálnom režime vidí adresáre a podadresáre s oddielmi a manipuluje s nimi, tento typ prístupu a manipulácie je potrebné vykonávať opatrne. Ak je napríklad zvyčajný adresár vytvorený alebo presunutý do adresára s oddielmi procesom v reálnom režime, tento adresár nebude nikdy viditeľný pre procesy spustené vo virtuálnom režime.

Aj keď adresár s oddielmi vyzerá pre proces vo virtuálnom režime ako zvyčajný adresár, pre adresár s oddielmi existujú určité obmedzenia.

Hierarchia:

Existuje hierarchia adresárov a podadresárov s oddielmi.

Hierarchia adresárov a podadresárov s oddielmi sa riadi nasledujúcimi pravidlami:
- Adresár musí mať jeden zo štyroch typov:
  - bežný adresár
  - adresár s oddielmi
  - podadresár s oddielmi
  - podadresár podadresára s oddielmi
- Adresár nikdy nemôže mať viac ako jeden typ
- Rodič podadresára s oddielmi musí byť adresár s oddielmi
- Každý adresár-potomok podadresára s oddielmi musí byť podadresár podadresára s oddielmi
- Rodič podadresára podadresára s oddielmi musí byť podadresár s oddielmi

Výsledkom každého porušenia týchto pravidiel bude neplatný strom adresárov s oddielmi a nekonzistentný súborový systém, ktorého správanie nie je definované.

Pripájanie súborových systémov:

Adresár alebo podadresár s oddielmi môže byť bodom pripojenia, ale podadresár podadresára nemôže byť bodom pripojenia. Podobne, koreňový adresár súborového systému, ktorý sa práve pripája, môže byť adresárom alebo podadresárom s oddielmi, ale nemôže byť podadresárom podadresára s oddielmi.

Vytváranie a vymazávanie adresárov:

Keď je proces vo virtuálnom režime spustený v pod-podadresári s oddielmi, vytvorí príkaz **mkdir** bežný adresár. Ak je rovnaký proces v podadresári s oddielmi a vykoná príkaz **mkdir**, je automaticky vytvorený pod-podadresár s oddielmi. Prázdny adresár je možné, s ohľadom na obmedzenia MAC, MIC a DAC, vymazať.

Presúvanie adresárov:

Pri presúvaní adresárov sú použité obmedzenia MAC, MIC a DAC.

Bežný adresár je možné presunúť kamkoľvek. Ak je jeho nový rodičovský adresár podadresár s oddielmi stane sa tento bežný adresár po presunutí pod-podadresárom s oddielmi. V opačnom prípade ostane naďalej bežným adresárom. Ak je jeho nový rodič adresár s oddielmi a jeho názov sa bije s názvom potenciálneho podadresára s oddielmi, zlyhá akékoľvek neskoršie presmerovanie procesu vo virtuálnom režime do tohto potenciálneho podadresára s oddielmi.

adresár s oddielmi môže byť presunutý do iného bežného adresára a aj po presunutí naďalej ostane adresárom s oddielmi. Vnorené adresáre s oddielmi nie sú v systéme Dôveryhodný systém AIX podporované, pretože neposkytujú žiadne dodatočné výhody.

Podadresár s oddielmi môže byť presunutý len do adresára s oddielmi a aj po presunutí ostáva naďalej podadresárom s oddielmi. Presunutie podadresára s oddielmi do bežného adresára, do podadresára s oddielmi, alebo do pod-podadresára s oddielmi nie je povolené.

Pod-podadresár s oddielmi je možné presunúť kamkoľvek. Ak jeho novým rodičom je bežný adresár, adresár s oddielmi, alebo pod-podadresár s oddielmi, stáva sa bežným adresárom. Inak ostane naďalej pod-podadresárom s oddielmi.

Tabuľka 42. Prehľad presunutí adresárov

Presun adresára typu	**Do bežného adresára**	**Do adresára s oddielmi**	**Do podadresára s oddielmi**	**Do pod-podadresára s oddielmi**
Bežný	Povolené. Ostáva bežným adresárom.	Povolené[1]. Ostáva bežným adresárom.	Povolené[1]. Stane sa pod-podadresárom s oddielmi.	Povolené. Ostáva bežným adresárom.
S oddielmi	Povolené. Ostáva adresárom s oddielmi.	Povolené[1]. Ostáva adresárom s oddielmi.	Nepovolené.	Povolené. Ostáva adresárom s oddielmi.
Podadresár s oddielmi	Nepovolené.	Povolené. Ostáva podadresárom s oddielmi.	Nepovolené.	Nepovolené.
Pod-podadresár s oddielmi	Povolené. Stane sa bežným adresárom.	Povolené. Stane sa bežným adresárom.	Povolené. Ostáva pod-podadresárom s oddielmi.	Povolené. Stane sa bežným adresárom.

[1] Ak sa názov bije s názvom potenciálneho (ešte neexistujúceho) podadresára s oddielmi, zlyhá akékoľvek neskoršie presmerovanie procesu vo virtuálnom režime do tohto podadresára s oddielmi.

Zmena typu adresára:

Príkaz **pdset** môžete použiť na zmenu bežného adresára na typ adresára s oddielmi. Neexistuje žiadny príkaz pre zmenu adresára s oddielmi na bežný adresár.

Nahradenie čísel inode:

Keď pristúpite na podadresár s oddielmi a vyžaduje sa jeho číslo inode alebo číslo inode jeho rodičovského adresára s oddielmi (..), vráti sa číslo inode jeho rodičovského adresára s oddielmi alebo číslo inode rodiča jeho rodičovského adresára s oddielmi. Keď pristúpite na podadresár s oddielmi a vyžaduje sa číslo inode rodiča podadresára s oddielmi (..), vráti sa číslo inode jeho prarodiča adresára s oddielmi.

Príkazy pre adresáre s oddielmi:

Tieto príkazy sa vzťahujú na adresáre s oddielmi.

pdmkdir
Vytvorí adresáre s oddielmi.

pdrmdir
Odstráni adresáre a podadresáre s oddielmi.

pdlink Prepojí súbory medzi podadresármi s oddielmi.

pdset Nastaví adresáre ako adresáre s oddielmi.

pdmode
Vráti prístupový režim aktuálneho adresára.

Spustí príkaz s prístupovým režimom zadaného adresára.

Regulárny adresár, ktorý bol skonvertovaný na adresár s oddielmi je možné skonvertovať späť na regulárny adresár.

Kontrola systémovej bezpečnosti:

Za kontrolu stavu systémovej bezpečnosti zodpovedá ISSO. Systémová bezpečnosť sa musí skontrolovať hneď po inštalácii, v pravidelných intervaloch a tiež pri každom ohrození systémovej integrity.

Databázový adresár systémovej integrity uložený v súbore /etc/security/tsd/tsd.dat obsahuje bezpečnostné informácie o objektoch súborového systému, napríklad dôležité príkazy a systémové zariadenia. Po pridaní nového zariadenia alebo modifikácii bezpečnostných informácií súborov je potrebné túto databázu zaktualizovať. Bližšie informácie získate pomocou príkazu **trustchk**.

Príkaz **trustchk** porovná aktuálne nastavenia bezpečnosti súboru, adresára alebo zariadenia s príslušnou položkou v databáze systémovej integrity a odstráni všetky protirečenia bezpečnostných atribútov. Len užívateľ s oprávnením ISSO môže spúšťať príkaz **trustchk**.

Riadenie TTY:

Minimálne SL, maximálne SL a TL pre zariadenia tty sú definované v databáze ttys v súbore /etc/login.cfg. Bližšie informácie získate pomocou príkazu **chsec**.

Platné SL užívateľovho prihlásenia cez port TTY by malo byť v rozsahu definovanom pre tento port v tomto súbore. Ak je pre port TTY zadané TL s výnimkou NOTL, platné TL užívateľa musí byť rovnaké ako zadané TL.

Riadenie vymazávaní užívateľov:

Každý užívateľ vrátane užívateľov s oprávnením ISSO, SA a SO musí mať na prihlasovanie do systému návestia. Vymazanie užívateľa môže byť zadané v súbore /etc/security/user ako súčasť užívateľovej strofy. Atribúty **minsl**,

maxsl, **defsl**, **mintl**, **maxtl** a **deftl** zadávajú pre užívateľa minimálne SL, maximálne SL, predvolené SL, minimálne TL, maximálne TL a predvolené TL v uvedenom poradí. Ak sú tieto atribúty zadané v užívateľovej strofe, hodnoty zadané v predvolenej strofe súboru budú pridelené užívateľovi.

Len užívateľ s oprávnením ISSO môže modifikovať databázu vymazávania bezpečnosti. Vymazanie užívateľa môže byť vypísané pomocou príkazov **lsuser** a **lssec** a modifikované pomocou príkazov **chuser** a **chsec**.

Hodnota predvoleného SL musí byť nižšia ako hodnota maximálneho SL a vyššia ako hodnota minimálneho SL. Podobne aj hodnota predvoleného TL musí byť nižšia ako hodnota maximálneho TL a vyššia ako hodnota minimálneho TL.

Poznámka: Aby sa užívateľ mohol úspešne prihlásiť do systému, musí platiť vyššie uvedený vzťah.

Správa systému pre administrátorov systému:

Užívatelia SA sú primárne zodpovední za aspekty administrácie systému, ktoré nesúvisia s bezpečnosťou.

Zodpovednosti užívateľov SA sú tieto:

- Pridávanie, odstraňovanie a udržiavanie kont užívateľov
- Zdieľanie úlohy zabezpečenia internej integrity systémového softvéru a súborových systémov s užívateľom ISSO
- Vytváranie a udržiavanie súborových systémov. Toto zahŕňa plánovanie štruktúry diskov, rozdelenie diskov do oddielov a zmenu veľkostí oddielov diskov, alokovanie odkladacieho priestoru a priestoru pre systémové a užívateľské adresáre, monitorovanie využitia súborových systémov, detekciu a ošetrenie chybných blokov diskov a správu priestoru na súborových systémoch presúvaním, vymazaním, archivovaním alebo komprimáciou súborov a súborových systémov.
- Identifikácia a oznamovanie problémov so systémom analýzou údajov o chybách a testovaním komponentov systému, akými sú súborové systémy, systémová pamäť a zariadenia.

Riadenie užívateľských kont:

Úlohou užívateľa s oprávnením SA je pridávať do systému nových užívateľov. Úlohou užívateľa s oprávnením ISSO je povoliť novým užívateľom prihlásiť sa a spúšťať príkazy v systéme.

Informácie o pridávaní oprávnení užívateľským kontám nájdete v kapitole Riadenie systému pre Information System Security Officers.

Keď užívateľ s oprávnením SA pridá do systému užívateľa, užívateľ s oprávnením ISSO o tom musí byť informovaný, aby bolo možné nastaviť úvodné heslo pre nového užívateľa pre prístup do systému.

Ak bolo stanovené, že užívateľ už nemá do systému prístup, mal by byť ihneď odstránený. Odstránenie užívateľa môže vykonať len užívateľ s oprávnením SA. ID užívateľa odstráneného zo systému nie je možné opätovne použiť, pokiaľ nie je vrátené pôvodnému užívateľovi a potom sa môže použiť len pri obnove tohto užívateľa v systéme.

Informácie o vytváraní a modifikácii užívateľských kont získate pomocou príkazov **mkuser**, **rmuser**, **chuser** a **pwadm**

Riadenie tlačiarní:

Po tom, ako bola tlačiareň správne nainštalovaná, je kombinovanou akciou užívateľov SA a SO pridaná do systému. Užívateľ SO pridá tlačiareň do systému a užívateľ SA vytvorí rozsah označení SL tlačiarne. Užívateľ ISSO má oprávnenie vykonať obe tieto úlohy.

Rozsah SL tlačiarne nesmie byť nastavený skôr, než bude tlačiareň pridaná do systému. Na riadenie tlačiarní použite príkaz **smit**.

Poznámka: Tlačenie súborov PostScript a ASCII na základe označení je podporované len na tlačiarňach PostScript.

MAC prístup k tlačiarni je určený označením SL procesu, ktorý súbor tlačí. Toto SL je zobrazené na stránkach s banerom, s hlavičkou/pätičkou a s trailerom. Proces, ktorý využíva príkaz **lp**, musí mať k súboru, ktorý ide tlačiť, MAC, MIC a DAC prístup. V opačnom prípade príkaz **lp** požiadavku na tlač nevygeneruje.

Keď je tlačiareň odstránená zo systému, mal by byť zo systému ihneď vymazaný profil tejto tlačiarne. To môže urobiť len užívateľ s autorizáciou SO.

Správa súborových systémov:

Súborový systém sa skladá z adresárov, údajových súborov, spustiteľných súborov a zo špeciálnych súborov. Súborový systém môže byť trvalo umiestnený na rôznych veľkokapacitných pamäťových zariadeniach, ako napríklad jednotky pevných diskov a diskety.

Hoci súborové systémy môže vytvárať a udržiavať len SA užívateľ, súborové systémy môžu pripájať aj odpájať aj SA aj SO užívatelia.

Kontrola súborových systémov pomocou príkazu fsck:

Internú integritu súborového systému pravidelne kontrolujte pomocou príkazu **fsck**. Príkaz **fsck** sa musí spúšťať na nepripojených súborových systémoch. Príkaz **fsck** môže spustiť len SA užívateľ.

Príkaz **fsck** sa štandardne spúšťa interaktívne, pričom vyzýva užívateľa na vykonanie akcie, keď nájde osirotený súbor alebo adresár. Užívateľ si môže zvoliť vymazanie súboru alebo sa môže pokúsiť súbor obnoviť. Ak užívateľ zadá, že súbor by sa mal obnoviť, príkaz **fsck** sa pokúsi uložiť súbor do adresára /lost+found.

Keď sa príkaz **fsck** dokončí a obnovené súbory budú uložené do adresára /lost+found, ISSO užívateľ by mal súbory prezrieť, aby stanovil úroveň ich zabezpečenia. Odporúča sa, aby mal adresár /lost+found priradené **SYSTEM_HIGH** SL, aby sa zamedzilo bežným užívateľom pristupovať na obnovené súbory.

Bližšie informácie nájdete v príkaze **fsck**.

Správa systému pre užívateľov SO:

Užívatelia SO sú primárne zodpovední za bezpečnostné aspekty správy systému.

Správa súborových systémov:

Za správu súborových systémov sú zodpovední užívatelia SO (System Officer)

Podporované súborové systémy:

Dôveryhodný systém AIX podporuje všetky diskové súborové systémy.

V Dôveryhodný systém AIX sú všetky súborové systémy s výnimkou JFS2 podporované ako jednoúrovňové súborové systémy. Tieto súborové systémy môžete pripojiť do systému Dôveryhodný systém AIX, automaticky im budú pridelené návestia a ostatné bezpečnostné atribúty a budú podliehať bezpečnostným mechanizmom, ktoré Dôveryhodný systém AIX uplatňuje. Všetky objekty súborov v jednoúrovňovom súborovom systéme majú rovnaké bezpečnostné atribúty zdedené z bodu pripojenia.

JFS2 je implementovaný na Dôveryhodný systém AIX ako viacúrovňový súborový systém a každý objekt súboru vo viacúrovňovom súborovom systéme má svoje vlastné bezpečnostné atribúty (bezpečnostné návestia). Adresár JFS2 má napríklad nezávislé minimálne a maximálne návestia SL.

V jednoúrovňových súborových systémoch sú minimálne a maximálne návestia SL bodu pripojenia rovnaké a všetky adresáre a súbory pod týmto bodom pripojenia sa tiež musia rovnať týmto návestiam SL.

Pripojenie a odpojenie súborových systémov:

Užívateľ SO (s oprávnením **aix.fs.manage.mount**) môže pripojiť alebo odpojiť súborový systém. Príkaz **mount** používa názov špeciálneho súboru zariadenia a adresár pripojenia ako voľby.

Po pripojení viacúrovňových súborových systémov JFS2 je adresáru pripojenia pridelené návestie koreňa súborového systému. Vo viacúrovňovom súborovom systéme má každý súbor svoje vlastné návestie integrity a senzitivity. Po modifikácii súboru sa jeho návestie príslušným spôsobom zaktualizuje.

Riadenie tlačiarní:

Užívateľ SO môže pomocou príkazu **lpadmin** pridávať, odstraňovať a upravovať tlačiarne a vykonávať iné akcie na konfiguráciu tlačového podsystému. Užívateľ SA môže pomocou príkazu **lpadmin** pridávať alebo upravovať návestia Sensitivity Label (SL) pre tlačiarne používať príkazy enable a disable na zapnutie a vypnutie tlačiarní.

Podsystém tlačiarne:

Podsystém tlačiarne vykonáva mnohé úlohy týkajúce sa prevádzky tlačiarne.

Úlohy podsystému tlačiarne zahŕňajú:
- Správu tlačiarní a ich atribútov
- Prijímanie, ukladanie a plánovanie tlačových úloh užívateľa
- Plánovanie tlačových úloh pre viaceré tlačiarne
- Spúšťanie programov, ktoré tvoria rozhranie s tlačiarňami
- Sledovanie stavu tlačiarní a tlačových úloh
- Hlásenie vzniknutých problémov
- Obmedzenie tlačových úloh užívateľa na tie, ktoré spadajú do rozsahu SL tlačiarne.
- Obmedzenie prístupu do odovzdaných tlačových úloh užívateľa
- Obmedzenie prístupu do adresárov a súborov podpory tlačiarne
- Správne označovanie tlačového výstupu návestiami

Funkcie bezpečnosti tlačiarne:

Podsystém tlačiarne je v Dôveryhodný systém AIX upravený tak, aby zahŕňal niekoľko funkcií bezpečnosti.

Podsystém tlačiarne je chránený a je vo vlastníctve systémového ID **lp**. Bežní užívatelia tak nemajú prístup do súborov podpory tlačiarne a adresárov s výnimkou užívateľových vlastných odovzdaných tlačových úloh a špeciálnych súborov zariadenia tlačiarne.

Podsystém tlačiarne skontroluje, či užívateľom odovzdaná tlačová úloha spadá do rozsahu návestí SL tlačiarne. Táto kontrola sa vykoná po odovzdaní tlačovej úlohy užívateľom pomocou príkazu **lp** a pred vytlačením odovzdanej úlohy démonom **lpsched**. Administrátor by mal byť informovaný o bezpečnostných kontrolách podsystému tlačiarne v prípade odmietnutia užívateľovej tlačovej úlohy.

Stránky s identifikačnými údajmi tlačovej úlohy (banner pages) sa tlačia pre všetky tlačové úlohy. Takáto stránka obsahuje ľahko čitateľné SL tlačovej úlohy a vytlačí sa pred a za každou tlačovou úlohou. Užívatelia môžu tlačiť aj bez nich, ale ide o auditovateľnú akciu. Vždy si skontrolujte, či sú návestia hlavičky a päty na všetkých stránkach správne a či sú menšie ako návestia na stránke s identifikačnými údajmi tlačovej úlohy.

Poznámka: Administrátor riadkovej tlačiarne musí zadať rozsah návestí pre každú tlačiareň. Ak chcete prideliť tlačiarni návestie, spustite príkaz:

lpadmin -d *printer_name* **-J***label* **-L***label* Uvedený príkaz zabezpečí, že na tlačiarni sa vytlačia len informácie označené týmto *návestím*.

Prehľad príkazov tlačiarne:

Niektoré príkazy podsystému tlačiarne môže spúšťať ktorýkoľvek užívateľ, ale niektoré len užívateľ s oprávnením SO, SA alebo ISSO.

Nasledujúca tabuľka uvádza zoznam príkazov podsystému stolovej tlačiarne, ktoré môžu spúšťať ľubovoľní užívatelia:

lp Zašle súbor do tlačiarne

lpstat Poskytne správu o stave podsystému tlačiarne

Príkazy správy podsystému tlačiarne vyžadujú oprávnenie SO s výnimkou užívateľa s oprávnením SA alebo ISSO, ktorý môže spúšťať príkaz **lpadmin** na zadanie rozsahu návestia tlačiarne a príkaz **lpstat** na zobrazenie tlačiarne a návestí SL požiadavky úlohy. Nasledujúca tabuľka vypisuje príkazy správy podsystému tlačiarne.

accept Povoľuje úlohy v tlačiarni

cancel Ruší tlačovú požiadavku súboru

disable Deaktivuje tlačiareň

enable Aktivuje tlačiareň

lpadmin
Nastaví alebo zmení konfiguráciu tlačiarne

lpfilter Nastaví alebo zmení filter tlačiarne

lpforms
Nastaví alebo zmení formulár tlačiarne

lpmove
Presunie tlačové požiadavky

lpsched
Vytlačí požiadavku

lpshut Zastaví tlačovú službu

lpusers
Nastaví alebo zmení prioritu tlače

reject Odmietne úlohy v tlačiarni

Riadenie tlačiarne z príkazového riadka:

Na riadenie tlačiarne z príkazového riadka sa používajú príkazy **accept**, **enable**, **disable**, **lpstat** a **lp**.

Príkaz **accept** sa používa na povolenie odoslania úloh do tlačiarne. Ak chcete povoliť, aby *laserová* tlačiareň prijímala tlačové úlohy, spustite príkaz:

```
/usr/sbin/accept laser
```

Tlačiareň zadaná pomocou *laser* môže teraz prijímať požiadavky na tlačové úlohy. Tlačové úlohy však budú vytlačené až po zapnutí tlačiarne. Ak chcete zapnúť tlačiareň, spustite príkaz:

```
/usr/bin/enable laser
```

Príkazy **enable** a **disable** sú administratívne príkazy a môže ich spúšťať len užívateľ s oprávnením ISSO alebo SA.

Ak chcete potvrdiť správnosť nastavenia tlačiarne, spustite príkaz **lpstat**:

```
lpstat -p laser -l
```

Tento príkaz zobrazí dlhú správu o stave *laserovej* tlačiarne. Ak spustíte príkaz **lpstat** bez voľby **-l**, zobrazí sa kratšia správa o stave. Ak má užívateľ oprávnenie SA alebo ISSO a použije voľbu **-l**, vypíše sa aj rozsah SL tlačiarne.

Ak chcete určiť stav tlačovej požiadavky, spustite príkaz **lpstat**:

```
lpstat -o
```

Tento príkaz vypíše všetky tlačové požiadavky **lp**. Ak má užívateľ oprávnenie SA alebo ISSO, platné návestie SL a vymazanie každej požiadavky bude zaznamenané.

Ak chcete vytlačiť názov súboru, spustite nasledujúci príkaz **lp**:

```
lp -d laser filename
```

V opačnom prípade musíte pri spúšťaní príkazu **lp** zadať cieľ tlačovej úlohy.

Ak administrátor nastavil predvolenú cieľovú tlačiareň, voľba **-d** *destination_ptr* nie je potrebná. Ak chcete napríklad vytlačiť názov súboru na laserovej tlačiarni, zadajte nasledujúci príkaz **lp**:

```
lp filename
```

Riadenie vypnutia systému:

Užívateľ SO môže systém vypnúť opätovným zavedením alebo úplným zastavením.

Užívateľ SO môže na opätovné zavedenie či zastavenie systému alebo na zmenu stavu init použiť nasledujúce príkazy:

reboot Automaticky rebootuje systém

halt Zastaví všetky systémové operácie

vypnutie
Zastaví všetky systémové operácie

init Zmení systémový stav init

Zálohovanie a obnovenie súborov:

Zálohy bránia strate údajov v prípade zlyhania hardvéru alebo nechceného vymazania súboru. Zálohy by mali byť vykonávané pravidelne, pričom medzi úplnými zálohami by mali byť vykonávané inkrementálne zálohy.

Príkazy **backup** a **restore** obsahujú voľby na zadanie názvov zálohových súborov, umiestnení, typov a ďalších volieb. Na vytvorenie inštalovateľného obrazu Dôveryhodný systém AIX pre skupinu jednotiek rootvg do súboru alebo na zaveditelnú pásku môžete použiť príkaz **mksysb**. Tieto príkazy môžete spustiť pomocou príkazu **smit**. Zálohy súborových systémov by mali vyť správne označené a uložené na bezpečnom mieste.

Programovanie Dôveryhodný systém AIX

Systémová bezpečnosť závisí od softvéru, hardvéru a firmvéru dôveryhodnej výpočtovej bázy (TCB) a zahŕňa celé jadro operačného systému, všetky ovládače zariadenia a moduly System V STREAMS, rozšírenia jadra a všetky dôveryhodné programy. Všetky súbory používané týmito programami sa pri rozhodovaní o bezpečnosti tiež považujú za súčasť TCB.

Vytváranie dôveryhodného softvéru vyžaduje dôkladné pochopenie základných princípov a funkcií systémovej bezpečnosti. Takmer všetky problémy v systémoch založených na UNIX vznikajú kvôli zle napísanému dôveryhodnému softvéru. Avšak pri kontrole bezpečnosti jadra Dôveryhodný systém AIX môžete napísať aplikácie používajúce rozšírené funkcie bezpečnosti. Aplikácia napísaná pre Dôveryhodný systém AIX môže rozlišovať súbory a procesy na rôznych úrovniach bezpečnosti a môže sa správať odlišne v závislosti od úrovne súboru alebo procesu používaného aplikáciou. Takáto aplikácia je známa ako viacúrovňová (MLS) aplikácia.

Programátor dôveryhodného systému musí dôkladne ovládať bezpečnostné funkcie Dôveryhodný systém AIX a rozumieť všetkým novým volaniam do systému Dôveryhodný systém AIX a knižniciam a príkazom týkajúcim sa bezpečnosti. Tieto informácie sú určené programátorom, ktorí vytvárajú alebo modifikujú dôveryhodný softvér. Tieto informácie obsahujú návody, zásady a upozornenia pre modifikáciu a vytváranie dôveryhodného softvéru. Tento

materiál ponúka úvodné vysvetlenia pre niektoré metódy a zásady bezpečnosti; programátorom dôveryhodných systémov sa však odporúča, aby si prečítali aj iné materiály o zabezpečených systémoch.

Zásady dôveryhodného softvéru

Jestvuje niekoľko dôležitých zásad podieľajúcich sa na vytvorení a modifikovaní dôveryhodného softvéru, vrátane dôvery a privilégií, návrhu dôveryhodného softvéru, najnižšieho privilégia, programovacích konvencií a ochrany TCB.

Dôvera a privilégium:

Proces môže obchádzať základné obmedzenia bezpečnosti (MAC, MIC, DAC a ostatné obmedzené operácie) len vtedy, ak je primerane privilegovaný. Každý proces spustený s privilégiom alebo privilégiami sa nazýva privilegovaný proces a program spustený daným procesom sa volá privilegovaný (dôveryhodný) program.

Výraz privilégium znamená individuálny atribút, ktorý procesu umožňuje vykonávať bezpečnostnú operáciu. Dôveryhodný systém AIX identifikuje a zoskupuje niektoré bezpečnostné operácie a prideľuje každej operácii iné privilégium, čím sa účinne odstráni privilégium superužívateľa (alebo koreňa) zo základného systému. Privilégiá sú prideľované procesom a súborom spustiteľných programov.

Programy musia byť za nasledujúcich okolností dôveryhodné:

- Program je nakonfigurovaný alebo určený na spustenie ako privilegovaný proces. Týka sa to každého programu, ktorý má byť spustený privilegovaným procesom.
- Pri rozhodovaní o bezpečnosti sa na tento program spolieha ďalší dôveryhodný program. Ak sa napríklad ostatné programy pri rozhodovaní o bezpečnosti spoliehajú na údaje v citlivej databáze, program, ktorý mení túto databázu, musí byť dôveryhodný.

Je dôležité zabezpečiť, aby nedôveryhodné programy nikdy nespúšťali privilegované procesy. Existuje niekoľko spôsobov, ako tomu predchádzať:

- Nepovoľujte bežne privilegovaným procesom spúšťať nedôveryhodné programy. Upozornite napríklad užívateľov spúšťajúcich privilegované programy podobné prostrediu shell, aby nespúšťali nedôveryhodné programy v privilegovanom programe podobnom prostrediu shell.
- Nikdy nepovoľte vlastné, zdedené ani oprávnené privilégiá pre súbory nedôveryhodného spustiteľného programu.

Všetky časti jadra operačného systému vrátane ovládačov zariadenia, modulov STREAMS a rozšírení jadra musia byť dôveryhodné. Objekty údajov, napríklad súbory a fyzické zariadenia sa tiež považujú za dôveryhodné, ak obsahujú informácie, na ktoré sa pri rozhodnutiach o bezpečnosti spolieha dôveryhodný program.

Návrh dôveryhodného softvéru:

Proces vytvárania dôveryhodného softvéru sa ponáša na proces pre ľubovoľný kľúčový softvérový komponent. Vytvoreniu dôveryhodného softvéru by mal predchádzať riadiaci cyklus dôkladne pochopenej a zdokumentovanej špecifikácie, návrhu, implementácie, testovania a konfigurácie.

Najdôležitejšou stránkou návrhu dôveryhodného softvéru je identifikácia subjektov a objektov a definícia presných akcií bezpečnosti na príslušnej úrovni abstrakcie. Väčšina bezpečnostných politík predstavuje obmedzenia pre subjekty, objekty a akcie. Keď subjekty vyžadujú oprávnenie na čítanie, zmenu alebo vytváranie objektov, bezpečnostné politiky monitorujú a schvália alebo odmietnu tieto požiadavky.

Subjekty

Subjekt je zvyčajne reprezentovaný ID užívateľa a skupiny. Na tento účel sa zvyčajne používa platný ID užívateľa a/alebo skupiny daného procesu, aj keď v niektorých prípadoch môže byť vhodné použiť reálny ID užívateľa a/alebo skupiny.

Objekty

Objekt je ľubovoľná kolekcia údajov a prístup do nej by mal byť riadený. Vo väčšine prípadov objekty sú súbory. Aj keď je bežné, že dôveryhodné programy riadia prístup do logicky rozdielnych objektov v tom istom súbore, vo všeobecnosti je lepšie mapovať objekty do súborov na báze "jeden-jeden".

V niektorých prípadoch sa subjekt tiež môže považovať za objekt. Proces sa napríklad zvyčajne považuje za subjekt. Keď sa však jeden proces pokúsi ovplyvniť druhý, tento druhý proces sa vzhľadom na danú operáciu zvyčajne považuje za objekt.

Požiadavky

Požiadavky sú súbory akcií, ktoré dôveryhodný modul vykoná pre subjekt. Každá požiadavka musí byť jasne identifikovaná kvôli vstupom, možným výstupom a výsledkom požiadavky vrátane všetkých vedľajších účinkov. Presná identifikácia všetkých požiadaviek predstavuje dôležitý úvod pre definovanie bezpečnostnej politiky.

Bezpečnostná politika

Bezpečnostná politika obsahuje jednoduché príkazy, ktoré uvádzajú, kedy sa budú vykonávať požiadavky zahŕňajúce špecifické objekty pre zadané subjekty. Subjekty, objekty a požiadavky musia byť dôkladne zadefinované a bezpečnostná politika by mala byť stručná a jasná. Je dôležité zadať identitu požadujúceho zúčastneného subjektu a objektov pre účely auditu.

Najnižšie privilégium:

Princíp najnižšieho privilégia stanovuje, že softvérovým modulom by mali byť na dosiahnutie ich plánovaných úloh pridelené najnižšie potrebné možnosti.

Najnižšie privilégium zahŕňa princíp, že dôveryhodné programy by mali dobrovoľne obmedziť vlastné citlivé spôsobilosti, aby ich bolo možné použiť v minimálnom možnom rozsahu oblastí programu. Najnižšie privilégium pomáha znížiť poškodenia zavinené chybami softvéru, alebo nepredvídanými bočnými efektmi. Všetky dôveryhodné programy by mali byť navrhované s ohľadom na pravidlo najnižšieho privilégia.

Priraďovanie a odnímanie privilégií:

Jednou z dôveryhodných softvérových techník je, že program vykonáva všetky operácie, pre ktoré sa vyžaduje privilégium, na začiatku svojej činnosti a potom uvoľní privilégium počas zvyšného času svojej činnosti. Toto sa volá zátvorkovanie privilégia.

Pozor na nasledujúce hľadiská týkajúce sa používanie privilégií:

- Procesu každého užívateľa sa priradí množina maximálnych privilégií pri spustení procesu. Túto množinu privilégií môže privilegovaný užívateľ kedykoľvek zmenšiť, ale nikdy nie zväčšiť.
- Je to zodpovednosť spúšťaného procesu zväčšovať a zmenšovať privilégiá maximálnej množiny na efektívnu množinu pri vykonávaní privilegovaných operácií.
- Privilégiá procesov sa modifikujú, keď procesy spúšťajú spustiteľné súbory, ktoré majú vlastné neprázdne množiny privilégií. Bližšie informácie nájdete v príkaze **exec**.
- Procesom sa pri ich spustení prideľuje aj množina limitujúcich privilégií. S príslušnými privilégiami môžu procesy zväčšovať privilégiá v maximálnej množine až privilégiá v limitujúcej množine.

Krátkodobé zmeny návestia MAC:

Keď proces musí zmeniť svoje návestie MAC z jeho normálneho prevádzkového návestia, trvanie zmeny návestia musí byť čo najkratšie. Toto je možné dosiahnuť použitím knižničných rutín.

Bližšie informácie o týchto knižničných rutinách nájdete v “Systémové volania Trusted AIX” na strane 472.

Krátkodobé otvorenia citlivých súborov:

Citlivý súbor je súbor, napríklad tieňový súbor hesiel, ktorý obsahuje informácie, ktoré by mohli kompromitovať bezpečnosť systému. Keď sú citlivé súbory otvorené na čítanie alebo na zápis, mali by byť otvorené len na nevyhnutný čas.

Atribút **close-on-exec** deskriptora súboru by mal byť nastavený použitím systémového volania **fcntl**. To zabráni neautorizovaným procesom v zdedení deskriptorov otvoreného súboru cez systémové volanie **exec**.

Centralizácia citlivých operácií:

Citlivá operácia je operácia, ktorá vyžaduje privilégiá. Ak je citlivá operácia vykonaná neprivilegovaným procesom, môže narušiť bezpečnosť systému.

Citlivé operácie by mali byť obmedzené na rozdielne moduly (podrutiny alebo oddelené programy). Pri rozpísaní veľkého programu do oddelených programov môžu tieto programy vyžadovať nižšie, alebo žiadne privilégiá. To zníži pravdepodobnosť náhodného narušenia bezpečnosti systému.

Použitie platných koreňových adresárov:

Program môže byť pripojený k určitému adresárovému stromu nastavením platného koreňového adresára programu na základný adresár stromu (pomocou systémového volania príkazu **chroot**) a nastavením pracovného adresára programu vnútri toho istého stromu. V skutočnosti ide o mechanizmus najmenšieho privilégia, pretože obmedzuje súbory, do ktorých môže vstupovať aj privilegovaný proces, na súbory v strome. Tento mechanizmus je efektívny najmä vtedy, keď rodičovský (dôveryhodný) proces takto obmedzí dôveryhodné alebo nedôveryhodné procesy potomkov.

Kým zmena koreňových adresárov poskytuje ochranu pre súbory mimo nového koreňového stromu, môže to spôsobiť problém s bezpečnosťou. Ak nebude zmena koreňového adresára vykonaná opatrne, bezpečnosť nového koreňového stromu môže byť ohrozená. Táto situácia môže nastať pri spracúvaní spojovacieho programu runtime a zdieľaných objektov v novom koreňovom strome. Túto procedúru používajte opatrne a úsporne.

Použitie chránených podsystémov:

Chránené podsystémy poskytujú ochranu integrity pre špeciálne podsystémy. Podsystém je kolekcia programov a/alebo dátových súborov, ktoré sú vo vlastníctve toho istého ID užívateľa a/alebo skupiny a používajú sa na implementáciu špecifickej funkcie v systéme.

Podsystém môže zahŕňať programy setuid alebo setgid. Chránený podsystém je podsystém s ID užívateľa, ktorý je ID systémového užívateľa.

ID systémového užívateľa je ID užívateľa s hodnotou menšou alebo rovnajúcou sa číslu 127. Užívatelia sa nemôžu prihlásiť s ID systémového užívateľa. Použitie chránených podsystémov môže významnou mierou znížiť počet privilegovaných procesov.

Režimy minimálneho prístupu:

Dôveryhodné programy (vlastne všetky programy) by mali objekty otvárať len v prístupových režimoch čítania/zápisu, ktoré sú nevyhnutné. V zásade to znamená, že by objekt nikdy nemal byť otvorený na čítanie-a-zápis, ak stačí prístup na čítanie. V obzvlášť citlivých prípadoch by mal proces otvoriť len na čítanie na konkrétnom mieste, v ktorom je zápis potrebný.

Tieto metódy sú dôležité predovšetkým vtedy, keď program vytvára ďalšie procesy, keďže prevádzanie privilégií a ďalšie všeobecné schopnosti (napríklad otváranie pripojení k citlivým súborom) sú kľúčovým aspektom dôveryhodného návrhu softvéru. Privilégiá môžu prepísať všetky obmedzenia. Pri vytváraní nových príkazov, ktoré budú mať privilégiá, by mal byť použitý starostlivý návrh a zváženie.

Ďalšie konvencie dôveryhodného programovania:

Dôveryhodný systém AIX využíva mnoho ďalších konvencií dôveryhodného programovania.

Redundancia:

Redundancia je užitočnou technikou pre bezpečnostné systémy. Bezpečnosť je zriedka absolútna, alebo skôr je takmer vždy otázkou umiestnenia dostatočného množstva prekážok do cesty kohokoľvek, kto sa pokúša nesprávne pristupovať do systému.

Výhodou redundantných bezpečnostných kontrol je, že ak jedna kontrola zlyhá alebo je kompromisná, ďalšie kontroly môžu poskytnúť ochranu. Nevýhodou redundantných kontrol je, že všeobecné bezpečnostné kontroly sú oddelené alebo distribuované cez systém. Z tohto dôvodu zatiaľ čo redundantné kontroly môžu byť extrémne užitočné, musia byť starostlivo navrhnuté, dokumentované a udržiavané.

Opakovanie kontroly, ktorú vykonáva jadro:

Málokedy je vhodné, aby proces vykonával kontroly, ktoré môže vykonať jadro. Proces by napríklad nikdy nemal čítať označenie MAC súboru a vykonať povinnú kontrolu vlastných prístupových práv. Kontrolu by vždy, keď je to možné, malo vykonať jadro.

Existujú dva hlavné dôvody, prečo by kontroly malo vykonávať jadro.

- Operácie jadra sú vzhľadom na ostatné procesy jadrové, zatiaľ čo kontrola procesu môže z hľadiska efektívnosti kolidovať s inými procesmi.
- Dôležitejšie je, že s novými verziami jadra môže dochádzať k zmenám v presných algoritmoch. Je namáhavé sledovať takéto zmeny v algoritmoch, ktoré sú súčasťou softvéru koncového užívateľa.

Priama kontrola privilégií:

Programy by sa nemali pokúšať určiť, či boli vyvolané ako privilegované procesy (napríklad skúmaním svojho efektívneho vektora, alebo vektora maximálneho privilégia procesu). Miesto by programy mali predpokladať, že boli vyvolané ako privilegované tam, kde je to potrebné.

Ak program nie je privilegovaným procesom, privilegované systémové volanie zlyhá a program môže vykonať príslušnú akciu. Zvyčajne nie je efektívnym bezpečnostným opatrením, aby program sám odmietol vykonať určité operácie, pokiaľ nie je privilegovaný. Ak je program privilegovaný, je kontrola nezmyselná. Ak program nie je privilegovaný, nemôže program narobiť viac škody, než akýkoľvek iný neprivilegovaný proces.

Táto kontrola ale môže byť efektívne použitá ako pomoc pri náhodnom zneužití. Môže byť zobrazené zmysluplné chybové hlásenie prehlasujúce, že tento program mal byť privilegovaný, ale nie je.

Šírenie citlivých schopností:

Citlivá schopnosť je schopnosť dôveryhodného programu, ktorá by mohla kompromitovať bezpečnosť systému, ak by bola poskytnutá nedôveryhodnému programu.

Keď privilegovaný program prenáša svoje privilégiá alebo všeobecné schopnosti na iné programy cez rodinu systémových volaní **fork** a **exec**, treba byť opatrný. Systémové volania **exec** sú najdôležitejšie, keďže tieto prenášajú privilégiá z jedného programu na druhý. Systémové volanie **fork** vytvorí nový proces, ale privilégiá nového procesu sú identické s privilégiami rodiča. Primárnym nebezpečenstvom je to, že spustiteľný súbor programu nemusí byť dôveryhodný alebo môže byť zmenený nedôveryhodným programom. Mali by ste venovať pozornosť tomuto:

- Dôveryhodné programy by nemali prenášať otvorené pripojenia na objekty (primárne súbory) na dcérske procesy, ak potomok a jeho potomkovia nie sú dôveryhodní na správny prístup k súboru v režime, v ktorom je súbor otvorený. Najlepšie pre proces by bolo prenášať nové pripojenie na objekt, ktorého režimy sú viac obmedzujúce, ako tie, ktoré by ináč existovali.

- Dôveryhodný proces, ktorý beží s iným efektívnym koreňovým adresárom, ako je absolútny koreň, by si mal byť istý, že jeho dcérske procesy nebudú zmätené. Napríklad ak dcérsky program otvorí dôveryhodný súbor, akým je tieňový súbor hesiel, môže použiť absolútny názov cesty, lebo predpokladá, že jeho efektívny koreň je absolútny.
- Môžu nastať prípady, v ktorých dôveryhodný program potrebuje zaviesť viac obmedzujúci umask na svojich potomkov.
- Dcérske procesy zdedia mnoho atribútov procesu. Ak dôveryhodný program vie, že dcérsky proces je nedôveryhodný a má návestie MAC, ktoré nie je nadradené návestiu dôveryhodného procesu a tieto atribúty boli zdedené dôveryhodným programom z nedôveryhodného predka, potom tieto atribúty môžu byť zdrojom potenciálnych tajných kanálov.
- Buďte opatrní pri šírení privilégií použitím systémových volaní **fork** a **exec**. Pri systémovom volaní **fork** sa privilégiá rodičovského procesu stanú privilégiami dcérskeho procesu. Počas systémového volania **exec** sú privilégiá modifikované.

V extrémne citlivých situáciách môže dôveryhodný program preveriť riadenie prístupov na dôveryhodný súbor, aby pomohol zaistiť, že je súbor správne chránený pred modifikáciami nedôveryhodnými programami. Napríklad môže byť požadované, aby bol súbor vlastnený užívateľom root s tým, že vlastníkovi súboru je povolené nanajvýš oprávnenie na zápis typu DAC.

Prostredie platného koreňového adresára:

Dôveryhodné programy sa často spoliehajú na správne absolútne cesty. Program **login** sa napríklad spolieha na to, že je /etc/security/passwd správnym tieňovým súborom hesiel.

To sa vzťahuje nie len na súbory s údajmi, ale aj na spustiteľné programy dôveryhodných programov. Hoci nedôveryhodné programy nemôžu využívať systémové volanie **chroot** na priamu zmenu platného koreňového adresára programu, môžu nastať situácie, v ktorých TCB nedôveryhodným programom umožní, aby boli spustené v platnom koreňovom adresári. Ak môžu tieto nedôveryhodné programy spúšťať dôveryhodné programy, ktoré sa spoliehajú na absolútne názvy ciest, môže to potenciálne spôsobiť bezpečnostné problémy.

Autentifikácia pomocou reálnych a efektívnych ID:

Je možné, že dôveryhodné programy budú musieť použiť niekoľko rozličných užívateľských a skupinových ID, ktoré sú procesu priradené. Je dôležité pochopiť rozdiely medzi týmito ID a ich správne použitie.

Reálne užívateľské a skupinové ID

Reálne ID užívateľa a skupiny zvyčajne predstavuje prihlasovacia identita relácie prihlásenia, v ktorej bol proces vytvorený. V niektorých prípadoch môžu byť reálne ID (najmä reálne ID užívateľa) použité pri bezpečnostných rozhodnutiach. Jednou takou inštanciou je kontrola autentifikácie. Reálne ID užívateľov využívajú príkazy ako spôsob overenia identity. To môže byť užitočné najmä pri zámerne škodlivých, alebo neopatrných použitiach riadiacich bitov **setuid-on-exec**, alebo **setgid-on-exec**. Kontrola skutočných ID však nie je súčasťou štandardných postupov UNIX a mala by byť vykonávaná len ak je nevyhnutná. Všeobecnou zásadou systémov UNIX je, že pri prístupe a ostatných bezpečnostných kontrolách mali byť využívané efektívne ID. K odklonu od tejto akceptovanej zásady by nemalo dôjsť bez dôsledného zváženia a dokumentácie.

Efektívne užívateľské a skupinové ID

Efektívne užívateľské a skupinové ID by mali byť využívané pri všetkých rozhodnutiach riadenia prístupov (DAC a MAC). Hodnoty ID systémových užívateľov sa nachádzajú medzi 0 a 127. Hodnoty ID normálnych užívateľov sú 128 a vyššie.

Absolútne názvy ciest pre dôveryhodné príkazy:

Niektoré schémy preniknutia bezpečnosťou sa pokúšajú vytvoriť falošný dôveryhodný program a umiestniť ho do cesty vyhľadávania programu podobnému prostrediu shell, používanej administračným či dokonca bežným užívateľom. Napríklad, falošná kópia príkazu **passwd** sa dá použiť na zachytenie existujúceho alebo nového hesla užívateľa.

Správnym administračným postupom je odstrániť aktuálny pracovný adresár z cesty vyhľadávania, aby sa tomu predišlo. Môžu však existovať iné cesty vyhľadávania, ktoré nie sú nevyhnutne dôsledne chránené a bežní užívatelia musia mať povolené vkladať aktuálny pracovný adresár do svojej cesty vyhľadávania. Účinným protiopatrením je, aby sa dôveryhodný program vždy vyvolával pomocou absolútneho názvu cesty (napríklad /usr/bin/passwd). Samotný dôveryhodný program skontroluje argument svojho prvého vyvolania a názov vyvolania. Ak sa nepoužije príslušný absolútny názov cesty, dôveryhodný program sa odmietne spustiť. Dôveryhodný program by tiež mal zabezpečiť, aby nemal účinný koreňový adresár, ktorý je iný ako absolútny koreň.

Poznámka: Toto je účinné len do tej miery, ako sú užívatelia navyknutí zadávať absolútny názov cesty. Ak užívateľ neuvedomene namiesto toho použije relatívny názov cesty a vyvolá sa falošný program, schéma preniknutia bezpečnosťou sa neodvráti.

Vytváranie štruktúry stromu adresárov:

Stromy adresárov by mali byť starostlivo štruktúrované, aby vylepšili ochranu kľúčových súborov. Základné usmernenia sú, že by mal byť prístup vyhľadávania v adresároch tak obmedzený, ako je to len možné (napríklad umiestnenie všetkých verejne dostupných súborov do adresárov, ktoré sú blízko koreňového adresára súborového systému).

Dobrým nápadom je aj umiestniť citlivé adresáre čo najbližšie k absolútnemu koreňovému adresáru, keďže to znižuje počet medzistupňových adresárov, ktoré budú musieť byť chránené.

Súborové systémy len na čítanie:

Asi najlepšie štruktúrovanie adresárového stromu je také, kde dôveryhodné súbory, ktoré sú zriedka menené, sú umiestnené na ich vlastný súborový systém a pripojené len na čítanie. Toto virtuálne zabezpečí, že ich obsah nemôže byť modifikovaný počas bežnej prevádzky systému. Táto technika je často používaná pre veľké kolekcie spustiteľných súborov pre dôveryhodné programy.

Ak je potrebná modifikácia súboru, môže byť súborový systém opätovne pripojený ako zapisovateľný vo viac chránenom kontexte (napríklad v režime jediného užívateľa alebo na oddelenom, viac zabezpečenom počítači). Odporúča sa, aby boli po takýchto aktualizáciách použité programy na skenovanie správnej konfigurácie súborového systému (napríklad správnych návestí DAC, MIC a MAC).

Navyše informácie DAC, MIC a MAC nie je možné zmeniť na súborovom systéme, ktorý je len na čítanie. Keď je súborový systém správne nakonfigurovaný, malo by toto chrániť proti schémam bezpečnostných prienikov, ktoré sa pokúšajú zmeniť informácie DAC a/alebo návestia MIC a MAC.

Spracovanie hesiel:

Všeobecne nie je správne, aby iné programy, ako štandardné systémové pomocné programy, žiadali od užívateľa prihlasovacie heslo. Heslá sú extrémne citlivé informácie a ich spracovanie by malo byť pevne obmedzené na niekoľko existujúcich vysoko dôveryhodných systémových pomocných programov.

Pre určité dôveryhodné podsystémy môže byť vhodné implementovať ich vlastné špecifické heslá. Spoliehanie sa na takéto súkromné schémy hesiel však môže byť nebezpečné, keďže tieto nie sú tak bezpečné ako mechanizmy podporené systémom.

Ochrana TCB (Trusted Computing Base):

Súbory s prvkami TCB musia byť chránené pred modifikáciou a v niektorých prípadoch pred prezradením (prečítaním) nedôveryhodnými programami.

Ochrana pred modifikáciou a prezradením je mimoriadne dôležitá. Súbory, ktoré je potrebné chrániť, zahŕňajú:

- Všetky súbory obsahujúce údaje používané dôveryhodným programom pri rozhodovaní o bezpečnosti (napríklad tieňový súbor hesiel)
- Všetky súbory spustiteľného programu pre dôveryhodný program
- Pseudosúbory, ktoré umožňujú prístup do častí TCB (napríklad /dev/kmem).

Poznámka: Súbory inicializácie systému (súbory rc) musia byť zvlášť chránené ako súčasť TCB

Ochrana pred modifikáciou:

Ochrana proti neautorizovanej modifikácii je primárne dosiahnutá nastavením informácií DAC na vyhovujúcu hodnotu. Normálne by boli tieto súbory vlastnené ID systémového užívateľa s oprávnením na zápis, povoleným len pre vlastníka súboru.

MIC je navrhnutý na ochranu proti modifikácii ochranou integrity objektov. Umiestnením vysokého návestia MIC na súbor je procesom s nižším návestím MIC zamedzené v modifikovaní, vymazaní alebo premenovaní súboru. Toto je ideálnou metódou na zamedzenie v nežiadúcej modifikácii súborov.

V niektorých prípadoch môže byť na ochranu proti neoprávnenej modifikácii použitý MAC. Avšak MAC je navrhnutý len na ochranu proti odhaleniu (čítaniu) a nie je dobre prispôsobený na ochranu proti modifikácii. Základná politika MAC nebráni subjektom v modifikovaní objektov s vyšším návestím. Aj keď nemajú povolené priame zápisy do súborov, určité dôveryhodné podsystémy to môžu povoliť. Taktiež veľa dôveryhodných súborov, ako sú spustiteľné súbory programov, musia mať nízke návestie MAC, aby boli všeobecne prístupné. Preto nastavenie vysokého štítka MAC na súbor nie je vždy možné.

Proti modifikácii súborov môžu chrániť aj bezpečnostné príznaky súborov. Niektoré bezpečnostné príznaky súborov bránia modifikácii objektov aj privilegovaným subjektom. Ak je pre súbor nastavený bezpečnostný príznak súboru **FSF_TLIB**, môže byť súbor zmenený len vtedy, ak je systém v režime konfigurácie, za predpokladu že je zapnutý prepínač bezpečnosti jadra **trustedlib_enabled**. Ak chcete pre súbor nastaviť **FSF_TLIB**, musí mať proces privilégium **PV_TCB** v jeho EPS. Ďalší relevantný bezpečnostný príznak súboru je **FSF_APPEND**, ktorý bráni modifikácii predtým zapísaných údajov. Do súboru s nastaveným príznakom **FSF_APPEND** je možné údaje len pridávať. Toto môže byť užitočné pre aplikáciu, ktorá protokoluje záznamy do súboru.

Tieto príznaky sú pre súbory zvyčajne nastavené skôr integrátormi, nie pod riadením programu. Programátori by sa mali byť vedomí týchto príznakov a ich funkcií.

Ochrana pred odhalením:

Na ochranu súborov TCB pred oprávnením na čítanie môže byť použitý DAC a MAC. Návestia MAC na týchto súboroch musia presne zohľadňovať citlivosť informácií v týchto súboroch. Napríklad ak je určitý algoritmus dôverný, potom musí byť náležite nastavené návestie MAC na spustiteľnom súbore programu, ktorý daný algoritmus používa.

Akceptovateľnou praxou je nastavenie návestia MAC umele vysoko (teda vyššie, ako je skutočná úroveň dôvernosti údajov v súbore) za účelom ochrany údajov pred odhalením. Avšak takáto nadmerná klasifikácia by mala byť použitá zriedka.

V takmer všetkých prípadoch na to, aby bol samotný súbor adekvátne chránený, musí byť chránená celá reťaz adresárov od absolútneho koreňa. Ináč by mohol škodlivý program odpojiť časť reťaze adresárov a vytvoriť nový podstrom s falošnou kópiou súboru.

Napríklad predpokladajme, že je dôveryhodný súbor uložený na /A/B/foo. Kým súbor **foo** je chránený pred úpravami, adresár **B** nie je. Škodlivý nedôveryhodný program by mohol odstrániť odkaz v **B** na **foo** a vytvoriť nový súbor **foo** s falošnou kópiou starého súboru **foo**. Dôveryhodné programy, ktoré otvoria /A/B/foo, budú potom otvárať falošný súbor a budú nevedomky oklamané tak, aby používali jeho údaje.

Dôveryhodné programy sa pri prístupe na súbory TCB spoliehajú na správne názvy ciest. Z tohto dôvodu by mali byť súbory so symbolickými odkazmi, používané v názvoch ciest pre súbory TCB, chránené rovnako silne ako samotné súbory.

V niektorých prípadoch môže byť na ochranu proti neoprávnenému odhaleniu použitý MIC. Avšak MIC je primárne určený na ochranu proti modifikácii (zápisu) a nie je dobre pripravený na ochranu proti odhaleniu.

Operácie s návestím citlivosti:

Existujú smernice dôveryhodných programov pre situácie, ktoré zahŕňajú subjekty alebo objekty s rôznymi návestiami citlivosti.

Mali by ste byť oboznámený s formou návestia citlivosti a vzťahom nadradenosti medzi návestiami. Vyššie znamená byť nadradený a nižšie znamená byť podriadený, kým rozšíriť znamená zvýšiť klasifikáciu údajov na vyššie návestie a zúžiť znamená klasifikáciu údajov na nižšie návestie.

Základné obmedzenie MAC:

Základné obmedzenie MAC (mandatory access control) je, že nedôveryhodné subjekty nemôžu spôsobiť, aby boli údaje s označením citlivosti A označené B, pokiaľ B nie je väčšie, než A.

Toto základné obmedzenie MAC sa vzťahuje na všetky triedy údajov. Zahŕňa obmedzenia pri zmene označenia údajov (to znamená pri zmene označenia údajového kontajnera) a pri presune označených údajov medzi údajovými kontajnermi.

Na rozdielnych úrovniach systému (systémové volanie, pomocné programy systémových služieb, a pod.), je toto základné obmedzenie usporiadané do konkrétnejších skupín pravidiel, ale vždy s touto rovnakou základnou filozofiou, že údaje môžu byť vo väčšine prípadov prevedené na vyššiu úroveň. Prvou úrovňou rozšírenia je napríklad to, že procesy môžu otvoriť na čítanie ktorúkoľvek veľkú triedu objektov, ak je označenie procesu vyššie, než označenie objektu a otvoriť na zápis, ak je označenie objektu vyššie, než označenie procesu.

V prípade bežného súboru sú operácie zápisu ďalej obmedzené na súbory s rovnakým označením, ako označenie procesu. Pri adresároch a zariadeniach sú operácie zápisu povolené, ak je SL subjektu vyššie, než minimálne SL objektu a maximálne SL objektu SL je vyššie, než SL subjektu. Pri špeciálnych FIFO súboroch (pomenované dátovody) sú z dôvodov skrytých kanálov operácie čítania tiež obmedzené na špeciálne FIFO súbory s rovnakým označením, ako označenie procesu.

Hoci môžu údaje migrovať na vyššie označenie citlivosti, nie je táto schopnosť vyžadovaná pri daných objektoch a situáciách. Samotný operačný systém napríklad neumožní, aby neprivilegovaný proces otvoril na zápis súbor s vyšším označením, aj napriek tomu, že to základné obmedzenie MAC umožňuje. To, či je toto rozšírenie povolené nedôveryhodným subjektom, je otázkou návrhu a filozofie. V niektorých prípadoch je to užitočné, a v niektorých nie. Problémom pri priamych zápisoch do súborov s vyšším označením je, že proces nemôže tieto súbory čítať, a tak je zápis do súboru s vyšším označením menej, než užitočný. Jednoduchý pomocný program, ktorý po požiadavke nedôveryhodného subjektu zvýši označenie súboru, však môže byť akceptovateľným a užitočným pomocným programom.

Na úrovni systémového volania sú obmedzené len neprivilegované procesy. To znamená, že privilegované procesy nie sú týmto obmedzením viazané. Prakticky sú ale všetky služby, ktoré dôveryhodný systém vykonáva, navrhované pre nedôveryhodných užívateľov, a preto na úrovni užívateľ-služba toto obmedzenie prevláda.

Základné obmedzenie MAC je používané pri všetkých spôsoboch, pomocou ktorých majú nedôveryhodné programy právo prenášať údaje. Toto základné obmedzenie MAC je však často rozdelené na dva komponenty. Prvý komponent sa zaoberá tými funkciami operačného systému, ktoré sú určené na prenos údajov (alebo označovanie). Tieto funkcie zahŕňajú napríklad čítanie a zápis do súborov a medziprocesovú komunikáciu údajov. Druhý komponent sa zaoberá spôsobmi komunikácie, ktoré ako také nie sú plánované; tie nazývame skryté kanály. Pokiaľ ide o skryté kanály, základné obmedzenie MAC je takmer nemožné uplatňovať. Z toho dôvodu je existencia skrytých kanálov s nízkou rýchlosťou prenosu (napríklad 0.1 bitov za sekundu) povolená, aj keď len vtedy, ak je zmysluplným kompromisom voči ostatným faktorom.

Základné obmedzenie MAC je priamočiare a jednoduché a pre prácu s viacúrovňovými údajmi je len relatívne málo podrobných usmernení.

Viacúrovňové operácie:

Systémové volanie **sec_setplab** umožňuje, aby privilegovaný proces ľubovoľne menil svoje označenie procesu.

Keďže takmer všetky obmedzenia MAC a MIC na neprivilegovaných procesoch sú uplatňované aj na privilegovaných procesoch už existujúcich systémových volaní (to znamená tých, ktoré sú zadefinované v základnom operačnom systéme), musí sa privilegovaný proces, ktorý potrebuje vykonávať viacúrovňové operácie, silne spoliehať ma systémové volanie **sec_setplab**. Dôveryhodné programy by však mali sec_setplab() využívať len nasledujúcim spôsobom:

- Každé využitie systémového volania **sec_setplab** na vykonanie viacúrovňových operácií (napríklad otvorenie súborov s vyšším označením na čítanie) by malo byť uskutočnené prostredníctvom rutín knižnice, ktoré odrážajú sémantiku aktuálnej vysokoúrovňovej vykonávanej operácie a ktoré zakryjú použitie systémového volania **sec_setplab**.
- Jedinou výnimkou sú veľmi jednoduché zmeny označenia procesu, ktoré nie sú súčasťou väčšej viacúrovňovej operácie. Tieto jednoduché operácie môžu systémové volanie **sec_setplab** použiť priamo.

Tieto pokyny na využívanie systémového volania **sec_setplab** majú dva dôvody. Prvým je, že taká citlivá a potenciálne nebezpečná funkcia, ako systémové volanie **sec_setplab**, by mala byť použitá len dobre navrhnutým a štandardným spôsobom. Druhým dôvodom je, pri súčasnom vývoji štandardov dôveryhodných systémov môžu systémové volania nižšej úrovne podporovať rozličné mechanizmy viacúrovňových operácií.

Zapuzdrené vysokoúrovňové operácie v rutinách knižníc poskytujú vynikajúcu stúpajúcu kompatibilitu a prispôsobivosť vyvíjajúcim sa verziám operačných systémov a pomáhajú zabezpečiť prenosnosť medzi dôveryhodnými verziami systému UNIX.

Dôveryhodné systémy poskytujú základnú skupinu týchto rutín. Tieto rutiny by mali byť používané zakaždým, keď je to možné. Táto skupina rutín by mala byť postupne rozšírená s pribúdajúcimi verziami operačného systému. Programátor dôveryhodných systémov môže tiež vytvoriť takéto rutiny knižníc tam, kde je to potrebné.

Ďalšou výnimkou obmedzení MAC a MIC je použitie jedného, alebo viacerých dostupných privilégií MAC, alebo MIC pri obchádzaní obmedzení MAC, alebo MIC. Použitie ktoréhokoľvek z týchto privilégií by malo byť vykonané veľmi obozretne.

System V Interprocess Communication (IPC):

Mechanizmus IPC (mechanizmus medziprocesovej komunikácie) (fronty správ, semafory a zdieľaná pamäť) podliehajú obmedzeniam DAC, MIC a MAC. Zvyčajne neexistujú príkazy na vytvorenie a používanie objektov System V IPC.

Systémové volania týkajúce sa AIX IPC boli modifikované na viacúrovňové pre Dôveryhodný systém AIX. Tieto modifikované systémové volania sú:

- **msgget**
- **msgsnd**
- **msgrcv**

- **msgctl**
- **semget**
- **semop**
- **semctl**
- **shmget**
- **shmctl**
- **shmat**
- **shmdt**

Okrem toho boli do Dôveryhodný systém AIX pridané nasledujúce systémové volania určené výslovne na manipuláciu s atribútmi MAC objektov IPC:

sec_getmsgsec
Získať bezpečnostné atribúty frontov správ

sec_getsemsec
Získať bezpečnostné atribúty semaforov

sec_getshmsec
Získať bezpečnostné atribúty segmentov zdieľanej pamäte

sec_setmsglab
Nastaviť bezpečnostné atribúty frontov správ

sec_setsemlab
Nastaviť bezpečnostné atribúty semaforov

sec_setshmlab
Nastaviť bezpečnostné atribúty segmentov zdieľanej pamäte

Požiadavky privilégií pre procesy na manipuláciu s objektmi IPC nájdete v kapitole Prístup do objektov IPC. Príkaz **settxattr** sa používa na manipuláciu s atribútom IPC.

Implementačné vysoké a systémové vysoké návestia MIC a MAC:

Dôveryhodný proces často musí stanoviť návestie MAC, ktoré dominuje nad všetkými ostatnými návestiami v systéme. Existujú dve rôzne návestia MAC, ktoré môžete použiť, implementačné vysoké návestie MAC alebo systémové vysoké návestie MAC.

Implementačné vysoké návestie MAC je najvyššie návestie MAC, podporované produktom Dôveryhodný systém AIX. Je pravdepodobné, že toto návestie má hierarchické členenie a obsahuje kategórie, ktoré sa pre túto lokalitu nepoužívajú. Toto návestie sa ľahko vygeneruje, ale návestie sa musí používať opatrne. V tomto návestí by žiadny proces nemal vytvárať objekty.

Systémové vysoké návestie MAC je najvyššie návestie MAC, ktoré sa používa pre lokalitu. Definuje ho administrátor v súbore **LabelEncodings**.

Používanie systémového vysokého návestia MAC je menej efektívne, ale dôrazne sa odporúča, pretože administrátor môže správnym nastavením príslušného parametra v súbore **LabelEncodings** efektívne obmedziť akcie aj privilegovaných procesov.

MIC má analogické implementačné vysoké návestie a systémové vysoké návestie.

Prihlasovacie rozsahy užívateľa a systému:

Dôveryhodné programy vykonávajúce služby pre užívateľov môžu chcieť limitovať návestia MIC a MAC zúčastňujúce sa týchto operácií na hodnoty, pri ktorých sa užívateľ môže prihlásiť a/alebo na prihlasovacie návestia povolené pre celý systém.

Vymazávania pridelené užívateľom v systéme sú v **užívateľskom** databázovom súbore /etc/security/user a prístup do nich je možný pomocou rutín knižnice **getuserattr** a **getuserattrs**.

Dôveryhodný systém AIX umožňuje užívateľom prevádzku v systéme pri ľubovoľnom návestí, ktoré je uvedené v rozsahu systémovej akreditácie a ktoré je nižšie ako maximálne vymazanie užívateľa a vyššie ako minimálne vymazanie užívateľa. Všetky programy, ktoré umožňujú užívateľom prevádzku pri iných návestiach, by mali vždy skontrolovať, či je nové návestie pre daného užívateľa platné.

Predpokladajme napríklad, že pomocný program s názvom **upgrade** bol zadefinovaný na zvýšenie návestia MAC v súbore na požiadavke ľubovoľného užívateľa. Základné obmedzenie MAC požaduje, aby pomocný program **upgrade** akceptoval len súbory, ktorých návestie MAC je nižšie než návestie užívateľa. Za opatrné sa ďalej považuje (aj keď to základné obmedzenie MAC striktne nevyžaduje), aby bolo nové návestie také, pri ktorom sa môže užívateľ prihlásiť, čo zahŕňa obmedzenia rozsahu návestí pre užívateľa aj pre celý systém. Pomocný program **upgrade** použije pre tento účel obe rozhrania **sl_cmp** a **accredrange**.

Štruktúra stromu adresárov:

Systémové volania fungujú tak, že stromy adresárov, vytvorené neprivilegovanými procesmi, nasledujú neklesajúcu štruktúru návestí, kde návestie súboru je rovnaké ako návestie jeho rodičovského adresára alebo je v rámci rozsahu adresára s oddielmi a návestie adresára je nadradené návestiu rodičovského adresára (nezabudnite že nadradenosť zahŕňa rovnosť). Toto je prirodzená štruktúra pre nedôveryhodné programy.

Avšak privilegované procesy nie sú viazané týmto obmedzením a môžu vytvárať stromy adresárov s ľubovoľnými vzťahmi návestí MAC rodičovského adresára. Takéto konfigurácie sú užitočné, lebo lebo prístup MAC na vyhľadávanie je obmedzený bližšie ku koreňu stromu. Napríklad ochrana agregácie, kde návestie MAC kolekcie údajových objektov je vyššie, ako ktorékoľvek samostatné návestie objektov, môže byť implementované nastavením návestia MAC adresára vyššie, ako má ktorýkoľvek z jeho elementov. Nedôveryhodné procesy potom musia ovládať návestie adresára, aby získali prístup k agregácii údajov.

Pri vytváraní stromov adresárov s klesajúcimi návestiami treba byť veľmi opatrný. Neprivilegovaný proces nedokáže otvoriť súbor na zápis, keď daný súbor nie je nadradený alebo rovný návestiu jeho rodiča.

Manipulácie s adresármi s oddielmi:

Je niekoľko systémových volaní, ktoré následkom implementácie adresárov s oddielmi prejavujú odlišné správanie.

Nasledujúce systémové volania sa po implementácii adresárov s oddielmi správajú odlišne:
- getdirents
- link
- mkdir
- mount
- rename
- rmdir
- stat
- lstat
- fstat

Režim procesu:

Príkaz **pdmode** môže spustiť príkaz so zadaným režimom. Proces môže použiť systémové volanie **setppdmode** na nastavenie svojho vlastného režimu na bežný alebo virtuálny režim. Systémové volanie **setppdmode** vyžaduje privilégium **PV_PROC_PDMODE**. Neexistuje žiaden mechanizmus na to, aby mohol proces zmeniť režim iného procesu.

Typ adresára:

Príkaz **pdset** môže byť použitý na zmenu normálneho adresára na adresár s oddielmi, ale neexistuje príkaz na zmenu adresára s oddielmi (alebo podadresára s oddielmi) na normálny adresár.

Na vytvorenie adresárov s oddielmi je možné použiť aj systémové volanie **pdmkdir**. Systémové volanie **pdmkdir** vyžaduje privilégium **PV_FS_PDMODE**.

Úvahy o návestí MIC a MAC:

Všetky programy by mali na zisťovanie vzťahu medzi návestiami MIC a MAC používať len funkcie **sl_cmp** a **tl_cmp**.

Toto je veľmi dôležité, lebo interný formát návestia sa môže v novších verziách systému zmeniť a tieto knižničné rutiny sledujú vyvíjajúce sa formáty. Rovnako existuje mnoho iných knižničných rutín, ktoré manipulujú návestia MIC a MAC, ktoré by mali byť použité kedykoľvek je to možné.

Systémové volania **setea**, **lsetea** a **fsetea** zmenia návestie MIC alebo MAC súboru. Systémové volanie **fsetea** akceptuje deskriptor súboru.

Ovládače zariadení:

Existuje niekoľko princípov a usmernení, ktoré by pri vytváraní ovládačov zariadení pre systémy Dôveryhodný systém AIX mali byť dodržané. Mali by ste byť oboznámený s mechanizmami vytvárania ovládačov zariadení pre základný systém a s predbežnými zákrokmi, ktoré s použitím týchto mechanizmov súvisia.

Podsystém riadenia zariadení:

Zariadenie v systéme AIX je abstraktný pojem a je využívané na pokrytie všetkých údajových objektov, ku ktorým je pristupované odkazmi špeciálnych súborov zariadení. V niektorých prípadoch sú tieto údajové objekty zastupované skutočnými fyzickými zariadeniami a niektorých prípadoch sú dosť odlišné (vrátane prípadov ako /dev/null, kde nie je vôbec žiaden objekt pamäte údajov). Druhé uvedené inštancie sa často nazývajú pseudozariadenia.

Systémy Dôveryhodný systém AIX poskytujú dva typy zariadení: zariadenia s jediným označením a viacúrovňové zariadenia. Viacúrovňové zariadenie je pre údaje procesu dôveryhodnými na viac než jednej úrovni citlivosti súčasne. Zariadenie s jediným označením je zvyčajne nedôveryhodné. Označenia na údajoch zvyčajne súvisia s informáciami, ktoré viacúrovňové zariadenie spracúva spôsobom, ktorý zabezpečuje že budú tieto údaje vždy správne označené. Zariadenie s jediným označením sa zvyčajne spolieha na externé označovanie.

Príkladom viacúrovňového zariadenia je pevný disk. Všetky údaje, ktoré sú umiestnené na pevnom disku, majú priradené označenia citlivosti. Tlačiareň, ktorá je fyzicky umiestnená v prostredí, ktoré pri vstupe vyžaduje platný bezpečnostný rozsah, je príkladom zariadenia s jediným označením. Tlačiarni je možné odoslať len údaje v platnom rozsahu označení.

Výstrahy vývoja ovládačov zariadení:

Ovládače zariadení sú súčasťou jadra operačného systému a ako také sú vo svojich akciách neobmedzené. Vytvorenie, alebo úprava ovládača zariadení, je citlivá ako úprava jadra samotného. Žiaľ, užívatelia musia vytvárať alebo upravovať ovládače zariadení. To by malo byť uskutočňované len s maximálnou obozretnosťou.

Je nemožné vymenovať zoznam všetkých konkrétnych výstrah, ktorých by ste sa pri písaní ovládača mali držať, pretože je priveľa spôsobov, akými môžu ovládače (niekedy celkom nevinne) úplne zničiť bezpečnosť systému. Preto je tvorba bezpečných ovládačov zariadení ponechaná skôr na posúdenie a skúsenosť ich návrhárov.

Ovládač zariadenia by nemal vykonávať nič iné, než jednoduché riadenie zariadenia. Pri ovládačoch zariadení, ktoré sú vytvorené v podstate len na pridávanie nových systémových volaní do systému, vrátane mnohých ovládačov pseudozariadení, ako napríklad tie pre /dev/kmem, by malo byť zvážené a následne navrhnuté nové volanie systému. Pokyny v tejto časti sa odkazujú najmä na tie ovládače, ktoré sú opodstatnenými správcami zariadení.

Predtým, než sa pokúsite vytvoriť nové ovládače zariadení, mali by ste študovať tie štandardné. Hlavné bezpečnostné akcie ovládača zariadení sú tie, ktoré sa týkajú spúšťania systémových volaní **open** a **ioctl**.

Otváranie zariadení:

Ako pri väčšine systémových objektov, je väčšina bezpečnostných kontrol súvisiacich s prístupom k zariadeniu vykonaná vtedy, keď je zariadenia otvárané systémovým volaním **open**.

Jadro najprv vykoná sadu základných operácií, a potom požiadavku na otvorenie posunie ovládaču zariadenia na spracovanie. Pred posunutím riadenia ovládaču zariadenia vykoná jadro nasledujúce kontroly bezpečnosti:

- Ak proces nemá k špeciálnemu súboru zariadenia oprávnenie MAC, proces otvárania zlyhá.
- Ak proces nemá k špeciálnemu súboru zariadenia oprávnenie MIC, proces otvárania zlyhá.
- Ak proces nemá k špeciálnemu súboru zariadenia oprávnenie DAC, proces otvárania zlyhá.

Pri mnohých zariadeniach môže čítanie zo zariadenia (systémovým volaním **read**) nahrádza stav zariadenia do určitej miery, ktorá je daná tým iným procesom, ktorý môže zistiť, že jeho označenie MAC nie je vyššie, než označenie čítajúceho procesu. To vytvára potenciálny skrytý kanál. Zariadenia, ktoré sú typu prvý-dnu-prvý-von (FIFO), vo svojej podstate predstavujú tento problém. V týchto prípadoch je bežnou praxou obmedziť oprávnenie na čítanie pre procesy, ktoré majú rovnaké označenie MAC, ako toto zariadenie. To je uskutočnené kontrolou v rámci ovládača zariadenia.

Existuje niekoľko špecifických pravidiel alebo pokynov pre návrh neštandardných zariadení. Musíte pochopiť a použiť základné princípy povinného a ľubovoľného riadenia prístupu. Našťastie môže byť väčšina ovládačov zariadení nakonfigurovaná ako bežné zariadenia a výstrednosti neštandardných zariadení nie je potrebné riešiť príliš často.

Príklady otvárania ovládačov zariadení:

Nasledujú príklady neštandardného ovládania zariadení prevzatého zo štandardných systémových ovládačov zariadení. Majú za úlohu vykresliť možné rozmanitosť takýchto ovládačov zariadení.

/dev/null

/dev/null je pseudozariadenie, ktoré nemá žiaden údajový kontajner. Údaje zapísané do /dev/null sú vymazávané a požiadavky na jeho čítanie zakaždým vrátia EOF (end-of-file). Preto nie je na toto zariadenie potrebné žiadne obmedzenie MAC. Pre porovnanie, prístup DAC je k súboru zariadenia /dev/null vyžadovaný, hoci to nie je prísne nevyhnutné.

/dev/tty

Keď proces vydá požiadavku otvorenia na /dev/tty, pokúsi sa ovládač zariadenia v skutočnosti otvoriť terminál, ktorý riadiacim terminálom žiadajúceho procesu. Preto musia byť skontrolované prístupy MIC, MAC a DAC k riadiacemu terminálu procesu, namiesto prístupov k /dev/tty. Pre porovnanie, prístup DAC k /dev/tty je vyžadovaný, hoci to nie je prísne nevyhnutné.

Obmedzenia rozhrania ioctl:

Hoci všetky funkcie rozhraní zariadenie-ovládač musia byť dôveryhodné, rozhranie **ioctl** si zvyčajne vyžaduje mimoriadnu pozornosť.

Je všeobecným pravidlom, že len procesy s oprávnením na zápis môžu nahradiť vlastnosť súboru, ktorá môže byť zistená iným procesom, ktorý nemá oprávnenie na zápis. Mať oprávnenie na zápis znamená buď to, že má proces tento súbor otvorený na zápis, alebo že je označenie MAC procesu rovnaké, ako označenie zariadenia. Toto obmedzenia vychádza zo základného obmedzenia MAC, že žiaden proces nemôže vykonať akciu, ktorú môžu zistiť procesy s nižšími označeniami MAC.

Ak dôvodom akcie je operácia čítania/zápisu užívateľských údajov, musí byť toto obmedzenie uplatnené tak, ako je uvedené. Za prípady, keď toto obmedzenie nie je uplatnené, naopak považujeme skryté kanály a mali by byť obmedzené z hľadiska šírky pásma a/alebo auditovateľné.

Niektoré akcie riadenia zariadení môžu byť na privilegované procesy obmedzené dokonca aj vtedy, ak toto zariadenie nie je nakonfigurované ako dôveryhodné zariadenie.

Iné obmedzenia:

Existuje relatívne málo ďalších prípadov, kedy by ovládač zariadenia musel uplatniť špeciálne bezpečnostné kontroly.

Jedným príkladom je, keď čítanie zariadenie nahrádza stav zariadenia do miery, akú môže zistiť proces, ktorého označenie MAC nie je vyššie, než označenie čítajúceho procesu. To predstavuje potenciálny skrytý kanál, ktorý by možno mal byť obmedzený, alebo auditovaný samotným ovládačom zariadenia.

Súhrn programovania ovládačov zariadení:

Pri implementácii ovládačov zariadení by mali byť zvážené nasledujúce usmernenia.

Poznámka: Boli pridané nové systémové volania na podporu rozšírenej bezpečnosti každého čítania/zápisu na zariadeniach Streams a FIFO. Tieto rozšírené bezpečnostné atribúty podporujú dve nové rozhrania API knižníc, eread() a ewrite(). V prípade jadra MLS je na zariadení nastavený bezpečnostná príznak DEV_SEC_ERDWR. Podobne je na zariadení FIFO nastavený príznak GNF_SEC_ERDWR. Tieto príznaky umožňujú pri každom čítaní/zápise dodatočné bezpečnostné kontroly.

Všeobecné metódy návrhov

Všetky bezpečnostné kontroly v rámci ovládača zariadenia by mali byť napísané modulárnym spôsobom a mali by byť jednoducho zistiteľné.

Kontroly v rámci ovládačov zariadení

Je vždy lepšie udržať kontroly MIC, MAC a DAC mimo ovládač zariadenia. Ovládače zariadení bez týchto kontrol môžu byť jednoducho presunuté z, alebo do dôveryhodného systému, alebo iných typov dôveryhodných systémov.

Pri implementácii štandardného ovládača zariadenia vykonáva jadro kontroly MIC, MAC a DAC a ovládač vykonáva akékoľvek ďalšie požadované kontroly privilégií. Pri implementácii neštandardného ovládača zariadenia sú všetky kontroly (kontroly MIC, MAC, DAC a privilégií) vykonávané v ovládači zariadenia. Rozhodnutie, či implementovať štandardný, alebo neštandardný ovládač zariadenia, je otázkou posúdenia návrhu.

DAC

DAC je uplatnený pre každý špeciálny súbor zariadenia, ktorý je na základe vstupného bodu súborového systému používaný na prístup k zariadeniu.

Kontrola správnej inštalácie

Každý ovládač zariadenia, ktorý vykonáva kontroly MAC, by mal (v rámci zmysluplných hraníc) bezpečne vyriešiť možnosť, že bolo zariadenie nesprávne zadefinované.

Privilegovaný prístup

Pre ovládač zariadenia môže nevhodné obmedziť určité operácie zariadenia len pre privilegované procesy. Aj napriek tomu existuje pre tieto situácie niekoľko špecifických odporúčaní.

Funkciou jadra **refmon** môžete určiť, či na to máte dostatočné privilégiá.

Najmenšie privilégium:

Dôveryhodný systém AIX uvádza koncept najmenšieho privilégia. Najmenšie privilégium oddeľuje kedysi výkonného užívateľa typu root do mechanizmu privilégií s jemnejšou štruktúrou. Toto rozdelenie privilégií zaručuje, že ak sa v dôveryhodnom softvéri vyskytne programovacia chyba alebo iná porucha, bezpečnosť systému bude len nepatrne poškodená.

Operácie s privilégiami:

Existujú štyri vektory privilégií súvisiace s každým procesom: skutočný, maximálny, dedičný a obmedzujúci.

Maximálny vektor privilégia definuje horné ohraničenie privilégia, ktoré môže byť aktívne pre každý proces. Skutočný vektor privilégia vymedzuje privilégiá, ktoré sú skúmané, aby bolo privilégium určené. Všimnite si, že skutočné privilégium je vždy podmnožinou maximálnej množiny privilégií, ktoré je vždy podmnožinou obmedzujúcej skupiny privilégií. Obmedzujúca skupina privilégií určuje privilégiá, ktoré môže mať proces vo svoje skupine maximálnych, dedičných, a skutočných privilégií. Skupina dedičných privilégií predstavuje skupinu privilégií, ktoré dedia potomkovia procesu cez rozvetvenia a vykonania.

Keď je vykonaný nový obraz textu, je eskalácia privilégií vykonaná na základe nasledujúceho algoritmu. Špeciálne spomenuté privilégiá sú **PV_ROOT**, **PV_SU_**, **PV_SU_EMUL**, **PV_SU_ROOT**, **PV_AZ_ROOT** a **PV_SU_UID**.

Nasledujúci algoritmus ukazuje dva dôležité koncepty podsystému najnižšieho privilégia. Prvý koncept je, že špeciálne privilégiá (**PV_ROOT**, **PV_SU_**, **PV_SU_EMUL**, **PV_SU_ROOT**, **PV_AZ_ROOT** a **PV_SU_UID**) sú jedinými privilégiami, ktoré sa môžu bezpodmienečne šíriť cez vykonanie nového obrazu procesu. Druhý koncept je, že vektor skutočného privilégia procesu je očistený o všetky privilégiá, až kým nie je nastavený súbor **FSF_EPS**. To zabezpečí spätnú kompatibilitu s aplikáciami, ktoré možno budú musieť byť spúšťané v dôveryhodnom systéme, bez toho, aby boli ohraničené pre systém najnižšieho privilégia.

```
new_max_privs = old_inheritable_privs
new_max_privs = new_max_privs | file_innate_privs
IF (user was assigned some of authorizations in file PAS)
new_max_privs = new_max_privs | file_authorized_privs
new_max_privs = new_max_privs & old_limiting_privs
IF (old_max_privs contain one or more special privileges)
new_max_privs += same set of special privileges
IF (FSF_EPS is set for the executable)
new_eff_privs = new_max_privs
ELSE
new_eff_privs = old_inheritable_privs
IF (old_eff_privs contain one or more special privileges)
new_eff_privs += same set of special privileges
new_limiting_privs = old_limiting_privs
```

Priraďovanie a odnímanie privilégií:

Nasledujúce rutiny štandardnej systémovej knižnice ilustrujú, ako sa manipuluje s privilégiami v systéme. Tieto rutiny sú použiteľné len na privilegovaných programoch v systéme.

priv_raise
Zmení vektor efektívneho privilégia procesu pridaním (alebo zväčšením) špecifikovaného zoznamu privilégií. Zoznam privilégií musí byť vo vektore maximálneho privilégia procesu, inak sa bude signalizovať chyba.

priv_remove
Zmení vektor efektívneho a maximálneho privilégia procesu odstránením špecifikovaného zoznamu privilégií. Ak proces nemôže odstrániť efektívne alebo maximálne privilégiá, bude sa signalizovať chyba.

priv_lower
Zmení vektor efektívneho privilégia procesu odstránením (alebo zmenšením) špecifikovaného zoznamu privilégií. Ak proces nemôže zmenšiť efektívne alebo maximálne privilégiá, bude sa signalizovať chyba.

Každá z týchto rutín akceptuje čiarkami oddelený zoznam privilégií, ktorý je ukončený číslom **-1** (mínus jedna, neplatné číslo privilégia). Technika pre zväčšovanie a zmenšovanie privilégií okolo najmenšej časti kódu, ktorá môže vyžadovať tieto privilégiá, sa nazýva zátvorkovanie privilégií. Všetky dôveryhodné aplikácie by mali používať zátvorkovanie privilégií na zníženie pravdepodobnosti narušenia bezpečnosti nedostatočne navrhnutým alebo implementovaným softvérom.

setppriv
Zmení vektor efektívneho, maximálneho, dedičného a limitujúceho privilégia špecifikovaním množín privilégií. Ak odovzdané množiny privilégií nie sú platné alebo povolené, bude sa signalizovať chyba.

Autorizácie:

Autorizácia poskytuje užívateľom s konkrétnymi oprávneniami rozličné skupiny privilégií.

Príkaz, alebo pomocný program štandardne skontrolujú všetky príslušné oprávnenia na začiatku svojho vykonávania, a na základe nich potom nastaví vlastné privilégiá. Preto užívatelia so špecifickou autorizáciou dostanú pri každom vykonávanom príkaze odlišné sady privilégií podľa toho, ako je príkaz naprogramovaný.

Aby bolo zo samotného kódu možné odstrániť nešikovné nastavenia privilégií, poskytuje systém AIX sady autorizácií a sady privilégií mimo binárneho kódu. So skupinami privilegovaných autorizácií (PAS) a skupinami autorizovaných privilégií (APS) vykonáva nastavenie privilégií založené na autorizácii skôr systém, než samotný príkaz.

checkauths
Porovnáva zoznam prevzatých autorizácií s autorizáciami, ktoré sú priradené aktuálnemu procesu.

Bližšie informácie o kontrole oprávnení nájdete v dokumente “Autorizácie RBAC” na strane 82.

Auditovanie:

Dôveryhodný systém AIXobsahuje sadu príkazov určených na riadenie generovania a informácií protokolov auditu. Je málo pravdepodobné, že bude programátor dôveryhodných systémov potrebovať tieto programy upraviť, alebo k nim niečo pridať.

audit Riadi démona auditovania.

auditbin
Riadi súbory protokolov auditu.

auditselect
Zlúči a vyberie záznamy auditu zo súborov protokolov auditu.

auditpr
Zobrazí zvolené udalosti auditu v čitateľnej forme.

Primárna oblasť, v ktorej má audit pre programátora dôveryhodných systémov význam, sú udalosti auditu, ktoré generujú dôveryhodné programy. Väčšina dôveryhodných programov musí zadávať správy do systémových protokolov auditu.

Situácie na auditovanie:

Existuje niekoľko presných pokynov na určenie, ktoré situácie je potrebné zisťovať a auditovať dôveryhodným programom. V prvom rade je to záležitosť posúdenia a stratégie auditu. Základný systém rozdeľuje situácie na úspešné, zlyhania, prístupy k objektom a možné skryté kanály.

Úspešné operácie:

Je dôležité auditovať úspešné operácie, aby sa vytvorila základná história používania.

Napríklad, je dôležité, aby program na alokovanie zariadení zaznamenal, keď konkrétny užívateľ alokuje a uvoľní zariadenie. To umožní programu sledovať tok informácií cez systém a určiť zodpovednosť, ak sa neskôr zistí, že zariadenie bolo zneužívané. Na druhej strane niektoré filozofie auditovania sa minimálne zaujímajú o úspešné operácie, lebo takéto operácie boli uznané za legálne a správne z dôveryhodného softvéru.

Zlyhania:

Auditovanie zlyhaných operácií môže byť užitočné na zisťovanie užívateľov, ktorí sa pokúsili získať prístup k nepovoleným službám alebo údajom. Častý výskyt takýchto zlyhaní môže signalizovať zlovoľných (ak nie zvlášť šikovných) pracovníkov.

Základný systém sa delí do piatich kategórií:

- Zlyhania privilégia (pokus neprivilegovaného procesu o vykonanie akcie, ktorá nie je obmedzená na privilegované procesy)
- Zlyhania MAC (zlyhanie akcie z dôvodu, že táto akcia by porušila obmedzenia MAC)
- Zlyhania MIC (zlyhanie akcie z dôvodu, že táto akcia by porušila obmedzenia MIC)
- Zlyhania DAC (zlyhanie akcie z dôvodu, že táto akcia by porušila obmedzenia DAC)
- Ostatné zlyhania (napríklad pokus o prihlásenie s nesprávnym heslom)

Prístupy k objektu:

Je potrebné auditovať prístup k objektu na monitorovanie užívateľov, ktorí pristupovali k danému objektu (napríklad tieňovému súboru hesiel).

Potenciálne skryté kanály:

Auditovanie potenciálnych skrytých kanálov je dôležité, lebo skryté kanály sa dajú používať na odovzdávanie informácií medzi procesmi na rôznych návestiach MAC. Používanie potenciálnych skrytých kanálov neznamená, že tieto kanály boli používané na tento účel, len to, že je toto použitie možné.

Každá položka zapísaná auditovým systémom obsahuje príčinu pre položku auditu (úspech, zlyhanie MAC, zlyhanie MIC, zlyhanie DAC, zlyhanie privilégia, iné zlyhanie, prístup k objektu alebo potenciálny skrytý kanál). To zahrňuje jednak záznamy auditu zapísané samotným systémom, ako aj záznamy auditu zapísané užívateľskými programami.

Môže byť užitočné posúdiť, či užívateľ bol dôveryhodný (t.j. administrátor), ale neexistuje absolútna metóda určovania, či dôveryhodný alebo nedôveryhodný užívateľ vyžaduje hlbšie auditovanie. Napríklad, hoci pre administrátorov sa predpokladá, že sú dôveryhodní a tým ohľadom môžu vyžadovať slabšie auditovanie, ich úkony môžu byť ďalekosiahle a môže byť užitočné zaznamenávať úkony neautorizovaného administrátora. Bežní užívatelia vykonajú menej škôd a v tomto zmysle vyžadujú slabšie auditovanie, ale sú tiež menej dôveryhodní a teda môžu vyžadovať silnejšie auditovanie. Administrátori systému často používajú na ich úkony silnejšie auditovanie, aby dokázali ich nevinu v prípade narušenia bezpečnosti.

Nasledujúce udalosti by mali by auditovateľné:

- Úspešné operácie, zvlášť tie, ktoré sa týkajú prenosu informácií alebo zmeny parametrov riadenia prístupu
- Operácie, ktoré zlyhali z bezpečnostných príčin

- Operácie vykonané administrátormi, úspešné aj neúspešné
- Potenciálne používanie skrytých kanálov
- Operácie, ktoré pristupujú ku konkrétnemu objektu
- Úkony, ktoré ovplyvňujú následný obsah skutočnej stopy auditu

Informačné úrovne auditov:

Informácie auditu vysokej úrovne sú užitočnejšie ako informácie auditu nízkej úrovne. Dôveryhodné programy si udržiavajú hľadisko operácií vysokej úrovne a môžu vytvárať vynikajúce správy auditu.

Zaznamenanie len toho, že administrátor otvoril bezpečnostný súbor na zápis, je oveľa menej užitočné ako zaznamenanie skutočnej operácie vysokej úrovne, ktorá bola vykonaná na súbore (napríklad zaznamenanie, že administrátor vytvoril novú položku v súbore, vrátane kľúčových informácií pre túto novú položku). Odporúča sa, aby informácie auditu boli na čo najvyššej možnej úrovni.

Je lepšie zahrnúť informácie o jednej udalosti ako zahrnúť informácie o viacerých udalostiach. Základnou príčinou na rozdelenie výskytu auditu na viac ako jednu udalosť je, aby boli tieto samostatné výskyty selektívne povolené.

Triedy a udalosti auditu:

Každý dôveryhodný program musí zisťovať triedu auditu, typ triedy auditu a dôvod, ktorý použije, keď vydáva správy auditu pomocou systémového volania **auditlog**.

Každá udalosť auditu patrí do triedy auditu. Priradením udalostí do tried môžete efektívnejšie spracovávať veľké množstvá udalostí. Definície triedy auditu sú definované v súbore /etc/security/audit/config.

Trieda auditu sa používa na povolenie a zakázanie zaznamenávania udalostí. Ak je dôležité samostatne povoliť dve udalosti, tieto dve udalosti by sa nemali nachádzať v tej istej triede auditu. Všeobecne je však dobrým zvykom zoskupovať udalosti do tried. Za normálnych okolností si každý dôveryhodný program alebo množina súvisiacich dôveryhodných programov vyhradí názov triedy auditu (alebo v zriedkavom prípade tried auditu) pre vlastné požitie.

Systémové akcie, ktoré sú auditovateľné, sú definované ako udalosti auditu v súbore /etc/security/audit/events.

Skryté kanály:

Pre všetok dôveryhodný softvér sa predpokladá, že sa nebude zúčastňovať schémach skrytých kanálov. Okrem toho musí byť softvér navrhnutý tak, aby sa nedal zneužiť nedôveryhodným softvérom na využívanie skrytých kanálov. Táto časť definuje skryté kanály a obsahuje pokyny na ich zisťovanie a obmedzovanie.

Definícia skrytých kanálov:

Žiadny proces na návesti A nebude schopný vykonať akciu, ktorá je zistiteľná iným procesom na návesti B, okrem prípadu, kedy návestie B je dominantné voči návestiu A.

Táto definícia sa dá rozdeliť do dvoch situácií: priame údajové operácie a príležitostné operácie. Priame údajové operácie sú určené pre užívateľov ako priamy prostriedok ukladania alebo komunikácie užívateľských údajov, napríklad čítanie pošty alebo zapisovanie súborov. Tieto operácie musia absolútne dodržiavať základné obmedzenie MAC. Všetky ostatné operácie sú príležitostné operácie. Použitie príležitostnej operácie na odovzdanie údajov v rozpore so základným obmedzením MAC sa nazýva skrytý kanál.

Využívanie skrytého kanála vyžaduje dva nedôveryhodné procesy, ktoré sa nazývajú odosielateľ (na návesti X) a príjemca (na návesti Y). Predpokladá sa, že návestie MAC príjemcu nie je také dominantné ako odosielateľa (keby áno, tak tok údajov od odosielateľa k príjemcovi by bol legálnou aktualizáciou). Aby mohli odosielateľ a príjemca využívať tento kanál, používajú určité konvencie týkajúce sa používania dohovorených prostriedkov za účelom prenosu údajov v rozpore s MAC.

Jediným kritérium na skryté zneužívanie je, že návestie príjemcu u príjemcu nie je dominantné nad návestím odosielateľa a že obaja, odosielateľ aj príjemca, sú nedôveryhodní. Odosielateľ aj príjemca sa bežne využívajú v prospech toho istého užívateľa. Predpokladá sa, že samotný TCB podporuje základné obmedzenie MAC a neobsahuje žiadny kód, ktorý by narušoval obmedzenie zlovoľným používaním skrytých kanálov. (V skutočnosti majú privilegované procesy oveľa účinnejšie spôsoby porušovania MAC bez toho, aby sa museli uchyľovať ku skrytým kanálom.) Ide o schopnosť nedôveryhodných procesov zneužívať skryté kanály využívaním dôveryhodných programov, o ktoré majú záujem.

Vo všeobecnosti je potrebné skryté kanály vylúčiť zo systému. Sú však iné prípady, kedy sa bez prítomnosti skrytých kanálov neprijateľne obmedzia iné potreby systému (napríklad výkon, spoľahlivosť alebo kompatibilita).

Smernice pre šírku pásma:

Základný systém používa nasledujúce smernice na obmedzovanie skrytých kanálov na základne šírky pásma:

Viac ako 100 bitov/s
: Existencia týchto kanálov je zakázaná

0,1 až 100 bitov/s
: Kanály v tomto rozsahu môžu existovať v prípade absolútnej potreby, ale ich používanie sa zisťuje a audituje vždy, keď je to možné

Menej ako 0,1 bitov/s
: Kanály v tomto rozsahu môže existovať, kde je to potrebné, ale nie je žiadna špeciálna potreba zisťovať ich používanie

Dôrazne sa odporúča, aby všetky ďalšie programy TCB dodržiavali tieto smernice. Okrem toho dávajte pozor na to, že aj relatívne pomalé kanály s rýchlosťou 10 bitov/s môžu preniesť 4500 bajtov za hodinu, čo je dostatočne významné množstvo údajov na nelegálne zúženie. Preto sa cení potrebná každá snaha o obmedzenie skrytých kanálov na čo najmenšiu možnú šírku pásma.

Šírka pásma najskrytejších kanálov most sa zvyčajne zmenšuje aktivitami procesov iných ako procesy využívajúce kanál. Odporúča sa však nespoliehať sa na tento efekt pri obmedzovaní šírky pásma skrytých kanálov, lebo na všetkých systémoch sú obdobia nízkej aktivity.

Zisťovanie skrytých kanálov:

Zisťovanie skrytých kanálov je vo veľkej miere záležitosť dôkladnej analýzy a dizajnu. Existuje niekoľko konkrétnych pokynov na zisťovanie skrytých kanálov.

Modul pojmu sa vzťahuje na jednotku kódu TCB, ktorý zisťuje alebo obmedzuje používanie skrytého kanála, či už v jadre alebo v procese. Zisťovanie skrytých kanálov je primárne záležitosť určovanie, či nedôveryhodný proces (odosielateľ) na úrovni A môže používať modul na vykonanie akcie, ktorá je zistiteľná iným procesom (príjemcom) na úrovni B, keď úroveň B nie je dominantnejšia ako úroveň A.

Napríklad, bežným skrytým kanálom sú údaje zapísané do súboru dôveryhodným procesom v prospech nedôveryhodného užívateľa, keď návestie MAC súboru nie je dominantnejšie ako návestie MAC užívateľa.

Bolo navrhnutých relatívne málo metodológií na zisťovanie skrytých kanálov. Najvýznamnejším je SRM (Shared Resource Matrix). Popis tejto techniky nájdete v nasledujúcich zdrojoch:

- • 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.

Zisťovanie skrytých kanálov pomocou auditovania:

Schopnosť auditovať potenciálne používanie skrytého kanála môže byť efektívnym opatrením proti tejto hrozbe. Aby však bolo auditovanie užitočné, udalosť auditovania musí byť relatívne zriedkavá. Audit nie je príliš užitočný, ak pomer skutočných zneužití k náhodnému používaniu udalosti, ktorá spôsobuje audit, je nízky.

Obmedzovanie skrytých kanálov:

Najlepším spôsobom na obmedzovanie skrytých kanálov je jednoducho ich odstránenie.

V opačnom prípade je potrebné ich obmedziť podľa pokynov prejednávaných v dokumente Pokyny pre šírku pásma. Okrem toho, vždy keď je to možné a efektívne, je potrebné auditovať potenciálne použitie kanálov.

VO všeobecnosti je ťažké pre kód ovládača jadra alebo zariadenia obmedziť skryté kanály, lebo kód ovládača jadra a zariadenia je určený pre efektivitu a ich kanály majú väčšiu šírku pásma. Dôveryhodné procesy môžu jednoduchšie obmedziť skryté kanály.

Poznámka: Nie je žiadny dôvod obmedzovať používanie skrytých kanálov na tom istom návesti alebo keď príjemca dominuje nad odosielateľom. Preto väčšina modulov TCB môže zvýšiť výkon systému nezavedením žiadnych obmedzení v týchto prípadoch.

Kvóty na jedno návestie:

Mnoho skrytých kanálov zahrňuje používanie oblasti prostriedkov, ktorá sa zdieľa medzi procesmi na rôznych návestiach MAC. Tieto je možné efektívne obmedziť vytvorením samostatných oblastí prostriedkov pevnej veľkosti pre každé návestie MAC, takže proces môže modulovať len používanie prostriedku z oblasti pre svoje návestie MAC.

Časom sa môžu nevyužité prostriedky presunúť z jednej oblasti do inej na uspokojenie dynamickej požiadavky. Táto migrácia prostriedkov je sama skrytým kanálom, ale s jednou z menších šíriek pásma, ktorý sa dá jednoducho obmedziť.

Časové oneskorenia:

Jednou technikou na obmedzovanie skrytých kanálov je, že TCB zabezpečí, aby prešlo určité množstvo času, keď sa vykoná služba, kde existujú kanály. To môže byť také jednoduché, že modul sa ponechá v spánku špecifikovaný čas, ktorý sa dá vypočítať na základe množstva odovzdávaných informácií.

Pokiaľ sa to však nevykoná správne, časové oneskorenia sa často zmaria programami, ktoré zneužívajú skrytý kanál. Napríklad, zneužívajúce procesy môžu vytvoriť množstvo množín procesov odosielateľa/príjemcu. Zatiaľ čo TCB dokáže jednoducho obmedziť každú množinu na určitú šírku pásma pomocou techník oneskorovania, súhrnom všetkých množín je šírka pásma tohto jedného kanála.

Je lepšie, ak určitá služba TCB zaistí, aby sa časové oneskorenia aplikovali nejakým spôsobom na všetky procesy, ktoré by mohli používať túto službu.

Časové oneskorenia môžu byť užitočné pre obmedzenie, ale sú náchylné ku jednoduchým protiopatreniam zlovoľnými programami a je potrebné ich navrhovať opatrne.

Údajové obmedzenia:

Šírka pásma skrytého kanála sa dá zmenšiť nielen zväčšením času, ale aj zmenšením množstva informácií, ktoré sa vracajú. Programy, ktoré vracajú údaje ako sériu operácií, môžu často jednoducho vrátiť menej alebo menšie pakety informácií v rovnakom časovom rámci.

Približný čas:

Mnohé z techník na zneužívanie skrytých kanálov vyžadujú, aby zneužívacie procesy mali presný spôsob na meranie relatívneho alebo absolútneho času. Tieto kanály môžu niekedy byť obmedzené a nepovoľovať procesu presne určiť čas.

Zatiaľ čo je relatívne jednoduché zaistiť, aby služby TCB, ktoré vracajú informácie o čase, dodávali približný čas, procesy niekedy majú iné spôsoby merania plynutia času, napríklad počítaním vlastných inštrukcií. Takéto techniky na obmedzovanie kanálov je potrebné používať s opatrnosťou.

Výrobcovia hluku:

Šírka pásma väčšiny skrytých kanálov sa zvyčajne zmenší, niekedy drasticky, aktivitami procesov iných, ako sú tie, čo zneužívajú kanál. Je možné, hoci sa to neodporúča, vytvoriť dôveryhodné programy, ktorých účelom je zaistiť, aby vždy existovala určitá úroveň aktivity. Tieto sa niekedy volajú ako výrobcovia hluku.

Zatiaľ čo využitie výrobcov hluku môže byť konceptovo príťažlivé, zvyčajne je pre výrobcov hluku ťažké určiť, kedy by mali vytvárať hluk a kedy nie. Preto sa neodporúča používať túto techniku na obmedzovanie kanálov.

Reťaze U-T-U:

Môžu nastať situácie, kedy nedôveryhodný proces, **U1**, vyvolá privilegovaný dôveryhodný proces, **T**, ktorý potom vyvolá iný nedôveryhodný proces, **U2**, ktorý je na inom návesti ako **U1**. **U1** a **U2** reprezentujú nedôveryhodné procesy na rôznych návestiach MAC so špeciálnym potenciálom skrytého kanála z poverenia, že jeden je potomkom druhého. (V skutočnosti T a U môžu byť postupnosti dôveryhodných a/alebo nedôveryhodných procesov.) Túto situáciu nazývame reťaz U-T-U.

Dôveryhodné procesy musia zabezpečiť, aby informácie neboli odovzdávané medzi dvoma nedôveryhodnými procesmi v rozpore so základným princípom MAC, ktorý zahrňuje vylúčenie nepovolených priamych údajových operácií a tiež skrytých kanálov. Do úvahy by ste mali vziať:

- Deskriptory súborov nemôžu zostať otvorené, keď **U2** nemohol otvoriť súbor v režime čítania/zápisu, v ktorom je otvorený
- Premenné prostredia sa musia vymazať, ak **U2** nedominuje U1
- Pracovný adresár odovzdaný od **U1** cez **U2** môže vytvárať skrytý kanál (pravdepodobne malý), ak návestie **U2** nedominuje **U1**. Podobne, mnohé parametre procesov, ktoré sa automaticky dedia dcérskym procesom, by mohli vytvárať skrytý kanál.

Reťaze U-T-U sa dajú správne riadiť (t.j. skryté kanály je možné dostatočne obmedzovať). Avšak je to ťažké zaistiť a reťaziam U-T-U sa treba všeobecne vyhýbať. Pozor však, problémom je, že U2 nie je dôveryhodný - mohol by byť bezpečne dôveryhodný, ale neprivilegovaný.

Príklady skrytých kanálov:

Nasledujú príklady skrytých kanálov, ktoré by mohli existovať v moduloch vytvorených programátorom systému.

Príklad skrytého kanála tlačovej služby:

Toto je príklad skrytého kanála tlačovej služby.

Dôveryhodná služba riadkovej tlačiarne správne označí každú odovzdanú úlohu návestím MAC vyžadovaného procesu a udržiava toto návestie s úlohami vo fronte pre prípadné použitie pri tlači. Úlohy s relatívne dlhými názvami sú povolené.

Stavový program umožňuje užívateľovi vidieť všetky úlohy, ktoré sú vo fronte pre užívateľa, vrátane užívateľom priradeného názvu úlohy, bez ohľadu na návestie úlohy. Toto je možné používať ako skrytý kanál, lebo proces odosielateľa môže potom vytvoriť úlohy, ktorých názvy obsahujú údaje, ktoré sa majú skryte odovzdať príjemcom pracujúcim v prospech toho istého užívateľa.

Poznámka: Jediným kritérium na skryté zneužívanie je, že návestie príjemcu u príjemcu nie je dominantné nad návestím odosielateľa a že obaja, odosielateľ aj príjemca, sú nedôveryhodní. Odosielateľ aj príjemca budú obaja pracovať v prospech toho istého užívateľa.

Tento kanál sa zatvorí, čo umožní užívateľovi len zobrazovať úlohy dominované podľa aktuálneho návestia MAC užívateľa. To si vynúti dominanciu návestia MAC príjemcu nad návestím MAC odosielateľa a kanál sa dá používať len na legálnu aktualizáciu. Ako zdvorilosť by mohol stavový program poskytnúť užívateľovi správu "existuje iná úloha", keby existovali iné ako majorizované úlohy. To reprezentuje oveľa menší kanál s dobrou prevádzkovou príčinou na existenciu.

Poznámka: Auditovanie zisťovania úlohy vyššej úrovne môže byť užitočné, lebo toto zisťovanie bude pravdepodobne vzácne pri bežnej prevádzke.

Toto je bežný príklad skrytého kanála, kde sú viacúrovňové pomenované údajové objekty (v tomto prípade tlačové úlohy vo fronte) dostupné prostredníctvom procesov na rôznych návestiach MAC. Tento kanál sa efektívne odstráni aplikovaním návestia MAC objektu aj na názov. Atribúty iné ako názov, napríklad veľkosť, tiež môžu niesť skryté informácie.

Príklad oblastí prostriedkov:

Keď dôveryhodný program vykoná službu pre nedôveryhodného klienta, dôveryhodný program alokuje zadaný typ prostriedku (napríklad vyrovnávaciu pamäť) z oblasti prostriedkov, ktorá sa zdieľa medzi procesmi na rôznych návestiach MAC.

Jedným spôsobom, ako možno toto využiť ako skrytý kanál, je, že odosielateľ a príjemca sa dohodnú na alokovaní všetkých prostriedkov okrem jedného, snáď pomocou iných programov spustených na iných alebo rozdielnych návestiach MAC alebo pod inými alebo rozdielnymi ID užívateľa. Odosielateľ potom spôsobí, že jeden zvyšný prostriedok bude alokovaný alebo nealokovaný a príjemca to zistí tiež pokusmi o alokovanie prostriedku.

Toto je klasický príklad zdieľaného kanála prostriedku. To je možné obmedziť alokovaním oblastí prostriedkov na jedno návestie, ako je popísané vyššie. Dá sa to tiež zistiť auditovaním.

Príklad databázy:

Dôveryhodný databázový systém umožňuje užívateľským programom umiestňovať údaje do viacúrovňovej databázy. Priamy prístup je primerane ovládaný cez základné obmedzenia MAC.

Avšak čas, vyžadovaný na umiestnenie položky do databázy veľmi závisí od aktuálnej celkovej veľkosti databázy. Predo môže odosielateľ umiestňovať alebo odstraňovať položky a ovplyvňovať tak veľkosť databázy a príjemca môže jednoducho odmerať čas, ktorý zaberie umiestnenie položky a tak zistiť jej veľkosť. Tento kanál má pravdepodobne malú šírku pásma, pokiaľ je prístup do databázy efektívny.

Za účelom obmedzenia kanála sa dá zaviesť garantovaná minimálna prístupová doba. Časové oneskorenie môže byť pseudonáhodné, takže priemerný stratený čas sa skráti. Toto je však stále schéma časového oneskorenia a je potrebné ju implementovať dôkladne.

Jednoduché auditovanie všetkých prístupov nebude pravdepodobne efektívne, lebo bude ťažké zistiť zneužívanie kanála medzi mnohými nezlovoľnými použitiami databázy.

Príklady programovania:

Táto časť obsahuje niekoľko dôveryhodných príkladov programovania

Príklad kontroly privilégia dôveryhodného programu:

Toto je modulárna rutina pre dôveryhodný program na kontrolu, či volajúci proces má alebo nemá konkrétne privilégium.

```
#include <sys/priv.h>
#include <sys/secattr.h>

int
priv_check (int priv)
{
  /* bezpečnostné atribúty procesu */
  secattr_t secattr;

/* získanie bezpečnostných atribútov volajúceho procesu */
  if ( sec_getpsec(-1, &secattr;) != 0 )
  {
    return (-1);
/* chyba pri získavaní štruktúry cred atribútov procesu */
  }

  /*
   * návrat, či už je alebo nie je špecifikované priv
   * v množine privilégií maxima volajúceho procesu
   */
  return privbit_test(secattr.sc_maxpriv, priv);
}
```

Príklad zmeny návestia zmeny efektívnej citlivosti:

Tento program zmení návestie efektívnej citlivosti aktuálneho procesu na systémový vysoký.

Vo vlastnej množine privilégií programu sa vyžadujú nasledujúce privilégiá:

- **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; /* Systémový vysoký SL */
 secattr_t attr;
 char *clBuffer = NULL;

 /*
  * Získanie systémového vysokého a nízkeho SL.
  */
 if ((sec_getsyslab(NULL, &sl_syshi, NULL, NULL)) != 0 ) {
  fprintf (stderr, "Volanie do sec_getsyslab zlyhalo.\n");
```

```
 exit(ERROR);
}

/*
 * Inicializovať tento proces s initlabeldb() pre prístup
 * k systémovej predvolenej databáze návestí.
 */
priv_raise(PV_LAB_LEF , -1);
if (initlabeldb(NULL) != 0) {
 fprintf(stderr,  "Nebolo možné čítať databázu kódovaní návestí.\n");
 exit(ERROR);
}
priv_remove(PV_LAB_LEF, -1);

/*
 * Získanie rozsahu vyúčtovania procesu a efektívneho SL.
 */
priv_raise(PV_LAB_SLUG, PV_LAB_SL_SELF, -1);
if (sec_getpsec(-1, &attr) != 0) {
 fprintf (stderr, "Problém pri získavaní bezpečnostných atribútov systému Trusted AIX pre program.\n");
 exit(ERROR);
}

/* malloc pre maximálnu dĺžku návestia SL, ktoré je možné vytvoriť pre proces */
if((clBuffer = (char *) malloc(maxlen_cl())) == NULL) {
 perror("malloc");
 exit(ERROR);
}
/* Skonvertovanie binárneho efektívneho SL na čitateľné pre ľudí */
if (clbtohr(clBuffer, &attr.sc_sl, HR_LONG) != 0) {
 fprintf(stderr, "Nie je možné skonvertovať SL do formy čitateľnej pre ľudí.\n");
 exit(ERROR);
}
printf("Počiatočný efektívny SL programu = %s.\n",clBuffer);

/*
 * Nastavenie efektívneho SL procesu na systémový vysoký.
 * Proces nemusí mať svoj maximálny SL na systémovom vysokom,
 * takže ho tiež nastavte na systémový vysoký.
 */
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, "Problém pri nastavení efektívneho SL pre program.\n");
 exit(ERROR);
}

priv_lower(PV_LAB_SLUG, PV_LAB_SL_SELF, -1);

if (sec_getpsec(-1, &attr) != 0) {
 fprintf (stderr, "Problém pri získavaní bezpečnostných atribútov systému Trusted AIX pre program.\n");
 exit(ERROR);
}

/* Skonvertovanie binárneho efektívneho SL na čitateľné pre ľudí */
if (clbtohr(clBuffer, &attr.sc_sl, HR_LONG) != 0) {
 fprintf(stderr, "Nie je možné skonvertovať SL do formy čitateľnej pre ľudí.\n");
 exit(ERROR);
}
printf("Upravený efektívny SL programu = %s.\n",clBuffer);
return(SUCCESS);
}
```

Príklady nastavenia klasifikácií návestí citlivosti a porovnávania návestí citlivosti:

Toto je príklad nastavenia klasifikácií návestí citlivosti a používania rutín knižníc na porovnávanie medzi návestiami citlivosti.

Privilégium **PV_LAB_LEF** sa vyžaduje v množine privilégií servera proxy programu a v množine privilégií maxima volajúceho procesu.

```
#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)
{
/* Návestia citlivosti */
sl_t sl1, sl2;

/* reťazce na uchovanie názvov návestí */
char *slBuffer1 = NULL;
char *slBuffer2 = NULL;

if (argc != 3) {
fprintf(stderr, "Použitie: compare slabel1 slabel2\n");
exit(ERROR);
}
/*
* Inicializovať tento proces s initlabeldb() pre prístup
* k systémovej predvolenej databáze návestí.
*/
priv_raise(PV_LAB_LEF , -1);
if (initlabeldb(NULL) != 0) {
fprintf(stderr,  "Nebolo možné čítať databázu kódovaní návestí.\n");
exit(ERROR);
}
priv_remove(PV_LAB_LEF, -1);

/* Konverzia odovzdaného SL do binárneho formátu */
if (slhrtob(&sl1, argv[1]) != 0) {
fprintf(stderr, "Nie je možné skonvertovať %s do binárneho formátu.\n", argv[1]);
exit(ERROR);
}
if (slhrtob(&sl2, argv[2]) != 0) {
fprintf(stderr, "Nie je možné skonvertovať %s do binárneho formátu.\n", argv[2]);
exit(ERROR);
}

/* malloc pre maximálnu dĺžku návestia SL, ktoré je možné vytvoriť */
slBuffer1 = (char *) malloc(maxlen_sl());
slBuffer2 = (char *) malloc(maxlen_sl());

if ((slBuffer1 == NULL) || (slBuffer2 == NULL)) {
perror("malloc");
exit(ERROR);
}

/*
* Preklad návestia späť na (dlhý) formát čitateľný pre ľudí.
* Toto nie je nevyhnutný krok. Je uvedený ako príklad
* použitia slbtohr() API.
*/
if (slbtohr(slBuffer1, &sl1, HR_LONG) != 0) {
fprintf(stderr, "Nie je možné skonvertovať do formy čitateľnej pre ľudí.\n");
```

```
exit(ERROR);
}

if (slbtohr(slBuffer2, &sl2, HR_LONG) != 0) {
fprintf(stderr, "Nie je možné skonvertovať do formy čitateľnej pre ľudí.\n");
exit(ERROR);
}

/*
* Použitie sl_cmp() na porovnanie dominancie dvoch návestí.
*/
if (sl_cmp(&sl1, &sl2) == LAB_SAME) {
printf("label (%s) equals label (%s).\n",
slBuffer1, slBuffer2);
}
else if (sl_cmp(&sl1, &sl2) == LAB_DOM) {
printf("label (%s) dominates label (%s).\n",
slBuffer1, slBuffer2);
}
else if (sl_cmp(&sl2, &sl1) == LAB_DOM) {
printf("label (%s) dominates label (%s).\n",
slBuffer2, slBuffer1);
}
else {
printf("Dve návestia sú rozpojené.\n");
}

return (SUCCESS);
}
```

Príklad nastavenia informácií auditu:

Tento program získa a nastaví informácie auditu.

Vo vlastnej množine privilégií programu sa vyžadujú nasledujúce privilégiá:

- **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;
 /* *Získanie masky predbežnej voľby auditu procesu */
 priv_raise(PV_AU_ADMIN, -1);
 rc = auditproc(0, AUDIT_QEVENTS, buf, sizeof (buf));
 priv_lower(PV_AU_ADMIN, -1);
 if (rc)
  fprintf(stderr, "Nepodarilo sa získať informácie auditu\n");
        /* *Pridanie triedy auditu jadra do masky predbežnej voľby */
 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, "Nepodarilo sa nastaviť informácie auditu\n");
 /* *Nastavenie GID procesu na generovanie záznamu auditu */
```

```
 priv_raise(PV_DAC_GID, -1);
 rc = setgid(129);
 priv_lower(PV_DAC_GID, -1);
 if (rc)
  fprintf(stderr, "Nepodarilo sa nastaviť gid\n");
 exit(0);
}
```

Príklad klienta:

Tento program odošle dve správy na server, jednu pomocou štandardnej rutiny **write** a druhú pomocou rutiny **ewrite**.

Zabezpečená správa sa odošle ako SECRET. Pozor, nezabezpečenej správe odoslanej pomocou volania **write** sa priradí predvolená množina bezpečnostných atribútov, ktoré sú konfigurovateľné cez netrule.

Vo vlastnej množine privilégií programu sa vyžadujú nasledujúce privilégiá:

- **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,  "Použitie:%s:  ADDR  PORT\n",  argv[0]);

  exit(1);
 }
#ifdef  SECURE
 /*
  * *  Získanie prístupu k databáze kódovaní návestí
  *
  *     */
```

```
 priv_raise(PV_LAB_LEF,-1);
 l_init_result  =  initlabeldb(NULL);
 if  (  priv_remove(PV_LAB_LEF,  -1)  !=  0  )
 {
  fprintf(stderr,  "Zlyhanie privilégia\n");
  exit(1);
 }
 if  (  l_init_result  !=  0  )
 {
  fprintf(stderr,  "Nebolo možné čítať databázu kódovaní návestí\n");
  exit(0);
 }
#endif  /*SECURE*/
  /*
   * *  Vyplnenie štruktúry "serv_addr"  adresou
   * zo
   * *  servera, ku ktorému sa chceme pripojiť.
   * */
 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]));
 /*  Otvoriť soket TCP (soket pre internetový tok).  */
 if  (  (sockfd  =  socket(AF_INET,  SOCK_STREAM,  0))  <  0)
 {
  perror("tcpclient:  ");
  fprintf(stderr,  "klient:  Nedá sa otvoriť soket pre prúd\n");
  exit(0);
 }
 if  (  connect(sockfd,  (struct  sockaddr  *)  &serv_addr;,
    sizeof(serv_addr))  <  0  )
 {
  perror("tcpclient:  ");
  fprintf(stderr,  "klient:  Nedá sa pripojiť na server\n");
  exit(0);
 }
 /*
  * *  Odoslanie normálneho zápisu a na server, ktorému sa
  * *  priradia predvolené bezpečnostné atribúty
  * */
 strcpy(buf,  "Tento má predvolené default bezpečnostné atribúty.\n");
 if  (  write(sockfd,  buf,  strlen(buf)+1)  ==  -1  )
 {
  perror("tcpclient:  ");
  fprintf(stderr,  "chyba zápisu\n");
 }
#ifdef  SECURE
  strcpy(buf,  "Táto správa je v SECRET\n");
 /*  Nastavenie SL a CL  */
 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;
 /*  Toto volanie ewrite potrebuje PV_MAC_CL a PV_LAB_SLUG_STR  */
 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("tcpclient  call");
  fprintf(stderr,  "chyba ewrite\n");
 }
```

```
 fflush(stderr);
#endif  /*SECURE*/
 fprintf(stderr,  "ukončuje sa .....  \n");
 sleep(3);
 close(sockfd);
 exit(0);
}
```

Príklad servera:

Tento program funguje ako server a používa rutinu **eread** na prijímanie správ odosielaných na jeho port. Po úspešnom prijatí správy tento program vypíše bezpečnostné atribúty správy.

Vo vlastnej množine privilégií programu sa vyžadujú nasledujúce privilégiá (bez priradenia FSF_EPS secflags):

- **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,  "Použitie:%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,  "Zlyhanie privilégia\n");
  exit(1);
 }

 if  (l_init_result  !=  0)
 {
  fprintf(stderr,  "Nebolo možné čítať databázu kódovaní návestí\n");
  exit(1);
 }
#endif  /*  SECURE  */
```

```
/*  Otvorenie soketu TCP (soket internetového toku).  */
if  (  (sockfd  =  socket(AF_INET,  SOCK_STREAM,  0))  <  0)
{
 perror("tcpserver:  ");
 fprintf(stderr,  "server:  Nedá sa otvoriť soket pre prúd\n");
 exit(1);
}
/*Naviazanie našej lokálnej adresy tak, aby klient mohol odosielať nám*/
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("tcpserver:  ");
 fprintf(stderr,  "server:  Nedá sa naviazať lokálna adresa\n");
 exit(0);
}
listen(sockfd,  5);
for  (;;)
{
 /*
  * *  Počkať na pripojenie z procesu klienta.
  * */
 fprintf(stdout,  "Čaká sa na pripojenie z klienta\n");
 clilen  =  sizeof(cli_addr);
 newsockfd  =  eaccept(sockfd,  (struct  sockaddr  *)  &      cli_addr,
   &clilen;,  &seclab;);
 if  (  newsockfd  <  0  )
 {
  perror("tcpserver:  ");
  fprintf(stderr,  "server:  chyba akceptovania\n");
 }
 /*  Tlač SL  */
 if  (  slbtohr(label_str,  &seclab.sl;,  HR_SHORT)  !=  0  )
 {
  fprintf(stderr,"problém s konverziou sl na reťazec\n");
 }
 else
 {
  fprintf(stdout,  "sl  =  %s.\n",label_str);
 }
 /*  Tlač MIN  CLEARANCE  */
 if  (  slbtohr(label_str,  &seclab.sl_cl_min;,  HR_SHORT)  !=  0  )
 {
  fprintf(stderr,"problém s konverziou min. vyúčtovania na reťazec\n");
 }
 else
 {
  fprintf(stdout,  "sl_cl_min  =  %s.\n",label_str);
 }

 /*  Tlač MAX  CLEARANCE  */
 if  (  slbtohr(label_str,  &seclab.sl_cl_max;,  HR_SHORT)  !=  0  )
 {
  fprintf(stderr,"problém s konverziou max. vyúčtovania na reťazec\n");
 }
 else
 {
  fprintf(stdout,  "sl_cl_max  =  %s.\n",label_str);
 }
 if  (  (childpid  =  fork())  <  0  )
 {
  perror("tcpserver:  ");
  fprintf(stderr,  "server:  chyba vetvenia\n");
  exit(0);
```

```
  }
  else  if  (  childpid  ==  0  )  /*  dcérsky proces  */
  {
   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");
      /*  Tlač SL  */
      if  (  slbtohr(label_str,  &e_seclab.sl;,  HR_SHORT)  !=  0  )
      {
   fprintf(stderr,"problém s konverziou sl na reťazec\n");
       }
      else
      {
   fprintf(stdout,  "sl  =  %s.\n",label_str);
       }
       /*  Tlač MIN  CLEARANCE  */
       if  (  slbtohr(label_str,&e_seclab.sl_cl_min;,HR_SHORT)!=  0)
      {
   fprintf(stderr,"problem  converting  min  CL  to string\n");
       }
      else
      {
   fprintf(stdout,  "sl_cl_min  =  %s.\n",label_str);
       }
       /*  Tlač MAX  CLEARANCE  */
      if  (  slbtohr(label_str,&e_seclab.sl_cl_max;,HR_SHORT)  !=0)
      {
   fprintf(stderr,"problem  converting  max  CL  to  string\n");
       }
       else
              {
   fprintf(stdout,  "sl_cl_max  =  %s.\n",label_str);
        }
        fflush(stdout);
#else  /*  NOT  SECURE  */
        fprintf(stdout,  "Calling  read\n");
        if  (read(newsockfd,  buf,  sizeof(buf))  <  1)
      {
   if  (ret  ==  -1)
```

```
      fprintf(stderr,  "read  error\n");
   else
      fprintf(stderr,  "read  no  data\n");
   close(newsockfd);
   exit(ret);
        }
        fprintf(stdout,  "%s\n",  buf);
        fflush(stdout);
#endif  /*  NOT  SECURE  */
              }
 }
 /*  parent  process  */
 close(newsockfd);
        }
}
```

Bezpečnostné atribúty užívateľa a portu systému Trusted AIX:

Bezpečnostné atribúty užívateľa a portu sa používajú na získavanie atribútov vyúčtovania užívateľov a portov a porovnávanie atribútov vyúčtovania užívateľa voči portom.

V súbore **usersec.h** sú definované nasledujúce ďalšie atribúty pre Dôveryhodný systém AIX.

S_MINSL
Návestie vyúčtovania minimálnej citlivosti užívateľ. Zadajte SEC_CHAR

S_MAXSL
Návestie vyúčtovania maximálnej citlivosti užívateľa. Zadajte SEC_CHAR

S_DEFSL
Predvolené návestie citlivosti užívateľa. Zadajte SEC_CHAR

S_MINTL
Návestie vyúčtovania minimálnej integrity užívateľa. Zadajte SEC_CHAR.

S_MAXTL
Návestie vyúčtovania maximálnej integrity užívateľa. Zadajte SEC_CHAR.

S_DEFTL
Predvolené návestie integrity užívateľa. Zadajte SEC_CHAR

Pre porty sú platné nasledujúce atribúty.

S_MINSL
Návestie minimálnej citlivosti priradené portu. Zadajte SEC_CHAR.

S_MAXSL
Návestie maximálnej citlivosti priradené portu. Zadajte SEC_CHAR

S_TL Návestie integrity priradené portu. Zadajte SEC_CHAR

Nasledujúci príklad určuje, či sa užívateľ môže prihlásiť na špecifikovaný port.

```
#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, "Použitie: %s <meno_užívateľa> <názov_portu>\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,
    "Chyba pri získavaní atribútov pre užívateľa %s\n", meno užívateľa);
  exit (1);
 }

 if (clhrtob (&(userlab->minsl), attributes[0].attr_char)) {
```

```
 fprintf(stderr, "chyba konverzie minsl\n");
 exit (1);
}

if (clhrtob(&(userlab->maxsl), attributes[1].attr_char)) {
 fprintf(stderr, "chyba konverzie maxsl\n");
 exit (1);
}

if (clhrtob(&(userlab->defsl), attributes[2].attr_char)) {
 fprintf(stderr, "chyba konverzie defsl\n");
 exit (1);
}

if (tlhrtob(&(userlab->mintl), attributes[3].attr_char)) {
 fprintf(stderr, "chyba konverzie mintl\n");
 exit (1);
}

if (tlhrtob(&(userlab->maxtl), attributes[4].attr_char)) {
 fprintf(stderr, "chyba konverzie maxtl\n");
 exit (1);
}

if (tlhrtob(&(userlab->deftl), attributes[5].attr_char)) {
 fprintf(stderr, "chyba konverzie deftl\n");
 exit (1);
}

printf("Užívateľ %s má nasledujúce hodnoty vyúčtovania\n", meno užívateľa);
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 ("Chyba pri získavaní atribútov portu");
 exit(1);
}

if (slhrtob(&(portlab->minsl), val)) {
 fprintf(stderr, "chyba konverzie minsl portu\n");
 exit (1);
}

if ( (rc = getportattr(portname,S_MAXSL, (char*)&val;, SEC_CHAR)) != 0 ) {
 perror ("Chyba pri získavaní atribútov portu");
 exit(1);
}

if (slhrtob(&(portlab->maxsl), val)) {
 fprintf(stderr, "chyba konverzie maxsl portu\n");
 exit (1);
}

if ( (rc = getportattr(portname,S_TL, (char*)&val;, SEC_CHAR)) != 0 ) {
 perror ("Chyba pri získavaní atribútov portu");
}
```

```
  if (tlhrtob(&(portlab->tl), val)) {
   fprintf(stderr, "chyba konverzie tl portu\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("Predvolený SL užívateľa nedominuje minimálnemu SL pre tty \n");
   exit(1);
  }

  cmpres = sl_cmp(&(portlab->maxsl), &(usrlab->defsl));
  if (cmpres != LAB_DOM && cmpres != LAB_SAME) {
   printf("Predvolený SL užívateľa nie je dominovaný minimálnym SL pre tty \n");
   exit(1);
  }

  cmpres = tl_cmp(&(portlab->tl), &(usrlab->deftl));
  if (cmpres != LAB_SAME) {
   printf("Predvolený TL užívateľa nie je rovnaký ako TL pre tty \n");
   exit(1);
  }

  printf("Užívateľ sa môže prihlásiť na špecifikovaný port\n");
  return;

}
```

Systémové volania Trusted AIX:

K dispozícii sú systémové volania na manipuláciu s ďalšími funkciami systému Dôveryhodný systém AIX.

eaccept
Akceptuje pripojenie na soket

ebind Vytvorí väzby na rozšírené položky na spracovanie bezpečnostných atribútov

econnect
Iniciuje pripojenie na rozšírený soket na spracovanie bezpečnostných atribútov

eread Prečíta z toku a získa bezpečnostné atribúty správy

ereadv Prečíta z toku a získa bezpečnostné atribúty správy

erecv Rozšírené recv, recvfrom, recvmsg na spracovanie bezpečnostných atribútov

erecvfrom
Rozšírené recv, recvfrom, recvmsg na spracovanie bezpečnostných atribútov

erecvmsg
Rozšírené recv, recvfrom, recvmsg na spracovanie bezpečnostných atribútov

esend Rozšírené send, sendto, sendmsg na spracovanie bezpečnostných atribútov

esendmsg
Rozšírené send, sendto, sendmsg na spracovanie bezpečnostných atribútov

esendto
Rozšírené send, sendto, sendmsg na spracovanie bezpečnostných atribútov

ewrite Zapíše do toku a nastaví bezpečnostné atribúty správy

ewritev
Zapíše do toku a nastaví bezpečnostné atribúty správy

sec_getmsgsec
Získa bezpečnostné atribúty frontov správ

sec_getpsec
Získa bezpečnostné informácie priradené k procesu

sec_getrunmode
Získa režim činnosti jadra

sec_getsecconf
Vráti aktuálne príznaky konfigurácie bezpečnosti

sec_getsemsec
Získa bezpečnostné atribúty semaforov

sec_getshmsec
Získa bezpečnostné atribúty zdieľaných pamäťových segmentov

sec_getsyslab
Získa predvolené systémové návestia citlivosti

sec_gettlibbufsize
Získa položky cesty knižnice v jadre

sec_gettlibpath
Získa položky cesty knižnice v jadre

pdmkdir
Vytvorí/nastaví/zruší nastavenie adresára alebo podadresára s oddielmi

sec_setauditrange
Nastaví rozsah návestí systémového globálneho auditu

sec_setplab
Nastaví návestie efektívnej citlivosti, návestie minimálneho vyúčtovania, návestie maximálneho vyúčtovania a návestie integrity špecifikovaného procesu

setppdmode
Nastaví režim adresára s oddielmi (reálny alebo virtuálny) procesu

setppriv
Nastaví množiny privilégií priradené k procesu

sec_setptlibmode
Nastaví režim TLIB procesu

sec_setrunmode
Nastaví režim činnosti jadra

sec_setsecconf
Nastaví príznaky konfigurácie bezpečnosti jadra

sec_setsemlab
Nastaví bezpečnostné atribúty semaforov

sec_setshmlab
Nastaví bezpečnostné atribúty zdieľaných pamäťových segmentov

sec_setsyslab
Nastaví návestia predvolenej citlivosti, informácií a integrity

Funkcie knižnice C systému AIX:

K dispozícii sú podprogramy a makrá na manipuláciu s ďalšími funkciami systému Dôveryhodný systém AIX.

accredrange
Určiť, či návestie citlivosti je v rozsahu akreditácie.

clbtohr
Skonvertovať dané binárne návestie vyúčtovania do formátu čitateľného pre ľudí

clhrtob
Skonvertovať dané vyúčtovanie čitateľné pre ľudí do binárneho formátu

getfsfbitindex, getfsfbitstring
Rutiny na získanie reťazcov a ukazovateľov príznaku bezpečnosti súboru

getmax_sl, getmax_tl
Získať návestia maximálnej citlivosti a integrity zo súboru kódovania návestí

getmin_sl, getmin_tl
Získať návestia minimálnej citlivosti a integrity zo súboru kódovania návestí

getsecconfig, setsecconfig
Rutiny na získanie a nastavenie konfiguračných príznakov bezpečnosti jadra pre režimy spúšťania

initlabeldb, endlabeldb
Rutiny na inicializáciu a ukončenie databázy návestí

maxlen_sl, maxlen_cl, maxlen_tl
Načítať maximálnu dĺžku návestí čitateľných pre ľudí na základe inicializovaného súboru kódovania návestí.

priv_isnull
Určí, či sú v danej množine privilégií zadané nejaké privilégiá

priv_lower
Operácie množiny privilégií

priv_raise
Operácie množiny privilégií

priv_remove
Operácie množiny privilégií

priv_subset
Operácie množiny privilégií

privbit_clr
Vymaže špecifikované privilégium v špecifikovanej množine privilégií

priv_clrall
Vymaže všetky privilégiá v špecifikovanej množine privilégií

priv_comb
Skombinuje prvé dve špecifikované množiny privilégií a výsledok umiestni do tretej špecifikovanej množiny privilégií

priv_copy
Skopíruje prvú špecifikovanú množinu privilégií do druhej špecifikovanej množiny privilégií

priv_isnull
Určí, či sú v danej množine privilégií zadané nejaké privilégiá

priv_mask
Vypočíta prienik prvých dvoch špecifikovaných množín privilégií a výsledok umiestni do tretej špecifikovanej množiny privilégií

priv_rem
Odstráni privilégiá v druhej špecifikovanej množine privilégií z prvej špecifikovanej množiny privilégií a výsledok umiestni do tretej špecifikovanej množiny privilégií

privbit_set
Nastaví špecifikované privilégium v špecifikovanej množine privilégií

priv_setall
Nastaví všetky privilégiá v špecifikovanej množine privilégií

priv_subset
Určí, či prvá špecifikovaná množina privilégií je podmnožina druhej špecifikovanej množiny privilégií

privbit_test
Otestuje, či špecifikovaná množina privilégií je nastavená v špecifikovanej množine privilégií

slbtohr, clbtohr, tlbtohr
Rutiny na konverziu binárneho návestia na čitateľné pre ľudí

slhrtob, clhrtob, tlhrtob
Rutiny na konverziu návestí čitateľného pre ľudí na binárneho návestia

sl_clr, tl_clr
Rutiny na vynulovanie návestí

sl_cmp, tl_cmp
Rutiny na porovnávanie návestí

tl_cmp Porovnať návestia integrity

Privilégiá systému Trusted AIX

V systéme Dôveryhodný systém AIX sú k dispozícii nasledujúce privilégiá. Poskytuje sa prehľad a opis každého privilégia a jeho použitie. Niektoré privilégiá tvoria hierarchiu, kde jedno privilégium môže udeľovať všetky práva súvisiace s ďalším privilégiom.

Pri kontrole privilégií systém najprv skontroluje a určí, či má daný proces najnižšie potrebné privilégium a potom v hierarchii skontroluje prítomnosť výkonnejších privilégií. Napríklad proces s privilégiom **PV_AU_** automaticky vlastní aj privilégiá **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE** a proces s privilégiom **PV_ROOT** automaticky vlastní všetky nasledujúce privilégiá s výnimkou privilégií **PV_SU_**.

Auditovacie privilégiá:

V systéme Dôveryhodný systém AIX sú dostupné nasledujúce auditovacie privilégiá. K dispozícii je syntéza a popis každého privilégia a jeho použitia. Niektoré privilégiá vytvárajú hierarchiu, v ktorej jedno privilégium môže udeľovať všetky práva súvisiace s iným privilégiom.

Pri kontrole privilégií systém najprv vykoná kontrolu, aby určil, či má proces najnižšie potrebné privilégium, a potom pokračuje smerom nahor v hierarchii, aby skontroloval prítomnosť výkonnejších privilégií. Napríklad proces s privilégiom **PV_AU_** má automaticky privilégium **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE** a proces s privilégiom **PV_ROOT** má automaticky všetky nižšie uvedené privilégiá, okrem **PV_SU_**.

PV_AU_
Ekvivalent k všetkým ostatným kombinovaným privilégiám **PV_AU_**.

PV_AU_ADD
Umožňuje procesu zaznamenávať/pridávať záznamy auditu.

PV_AU_ADMIN
Umožňuje procesu konfigurovať a dotazovať auditovací systém.

PV_AU_PROC
Umožňuje procesu získať a nastaviť stav auditu procesu.

PV_AU_READ
Umožňuje procesu čítať súbor označený ako súbor auditu.

PV_AU_WRITE
Umožňuje procesu zapisovať do, alebo vymazať súbor označený ako súbor auditu, alebo označiť niektorý súbor ako súbor auditu.

Autorizačné privilégiá:

V systéme Dôveryhodný systém AIX sú dostupné nasledujúce autorizačné privilégiá. K dispozícii je syntéza a popis každého privilégia a jeho použitia. Niektoré privilégiá vytvárajú hierarchiu, v ktorej jedno privilégium môže udeľovať všetky práva súvisiace s iným privilégiom.

Pri kontrole privilégií systém najprv vykoná kontrolu, aby určil, či má proces najnižšie potrebné privilégium, a potom pokračuje smerom nahor v hierarchii, aby skontroloval prítomnosť výkonnejších privilégií. Napríklad proces s privilégiom **PV_AU_** má automaticky privilégium **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE** a proces s privilégiom **PV_ROOT** má automaticky všetky nižšie uvedené privilégiá, okrem **PV_SU_**.

PV_AZ_ADMIN
Umožňuje procesu upravovať bezpečnostné tabuľky jadra.

PV_AZ_READ
Umožňuje procesu získať bezpečnostné tabuľky jadra.

PV_AZ_ROOT
Spôsobí, že proces úspešne prejde kontrolou autorizácie počas systémového volania **exec**.

PV_AZ_CHECK
Umožní procesu úspešne prejsť všetkými kontrolami autorizácie.

Privilégiá DAC:

V systéme Dôveryhodný systém AIX sú dostupné nasledujúce privilégiá DAC. K dispozícii je syntéza a popis každého privilégia a jeho použitia. Niektoré privilégiá vytvárajú hierarchiu, v ktorej jedno privilégium môže udeľovať všetky práva súvisiace s iným privilégiom.

Pri kontrole privilégií systém najprv vykoná kontrolu, aby určil, či má proces najnižšie potrebné privilégium, a potom pokračuje smerom nahor v hierarchii, aby skontroloval prítomnosť výkonnejších privilégií. Napríklad proces s privilégiom **PV_AU_** má automaticky privilégium **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE** a proces s privilégiom **PV_ROOT** má automaticky všetky nižšie uvedené privilégiá, okrem **PV_SU_**.

PV_DAC_
Ekvivalent k všetkým ostatným kombinovaným privilégiám PV_DAC_.

PV_DAC_O
Umožňuje procesu prepísať obmedzenia vlastníctva DAC.

PV_DAC_R
Umožňuje procesu prepísať obmedzenia čítania DAC.

PV_DAC_W
Umožňuje procesu prepísať obmedzenia zápisu DAC.

PV_DAC_X
Umožňuje procesu prepísať obmedzenie vykonávania DAC.

PV_DAC_UID
Umožňuje procesu nastaviť, alebo zmeniť ID (UID) jeho užívateľa.

PV_DAC_GID
Umožňuje procesu nastaviť, alebo zmeniť ID (GID) jeho skupiny.

PV_DAC_RID
Umožňuje procesu nastaviť, alebo zmeniť ID (RID) jeho roly.

Privilégiá súborového systému:

V systéme Dôveryhodný systém AIX sú dostupné nasledujúce privilégiá súborového systému. K dispozícii je syntéza a popis každého privilégia a jeho použitia. Niektoré privilégiá vytvárajú hierarchiu, v ktorej jedno privilégium môže udeľovať všetky práva súvisiace s iným privilégiom.

Pri kontrole privilégií systém najprv vykoná kontrolu, aby určil, či má proces najnižšie potrebné privilégium, a potom pokračuje smerom nahor v hierarchii, aby skontroloval prítomnosť výkonnejších privilégií. Napríklad proces s privilégiom **PV_AU_** má automaticky privilégium **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE** a proces s privilégiom **PV_ROOT** má automaticky všetky nižšie uvedené privilégiá, okrem **PV_SU_**.

PV_FS_
Ekvivalent k všetkým ostatným kombinovaným privilégiám **PV_FS_**.

PV_FS_MKNOD
Umožňuje procesu vykonať systémové volanie **mknod**, aby vytvoril súbor akéhokoľvek typu.

PV_FS_MOUNT
Umožňuje procesu pripojiť a odpojiť súborový systém.

PV_FS_CHOWN
Umožňuje súboru zmeniť vlastníctvo súboru.

PV_FS_QUOTA
Umožňuje súboru riadiť informácie súvisiace s kvótami disku.

PV_FS_LINKDIR
Umožňuje súboru vytvoriť pevné pripojenie k adresáru.

PV_FS_RESIZE
Umožňuje súboru vykonať na súborovom systéme zväčšujúce a zmenšujúce typy operácií.

PV_FS_CNTL
Umožňuje súboru vykonať na súborovom systéme rozličné operácie riadenia, okrem operácií zväčšovania a zmenšovania.

PV_FS_CHROOT
Umožňuje procesu zmeniť jeho koreňový adresár.

PV_FS_PDMODE
Umožňuje procesu vytvoriť, alebo nastaviť adresár typu s oddielmi.

Privilégiá procesu:

Na Dôveryhodný systém AIX sú k dispozícii tieto privilégiá procesu. Je poskytnutý prehľad a popis každého privilégia a jeho použitie. Niektoré privilégiá vytvárajú hierarchiu, kde jedno privilégium môže udeliť všetky práva priradené inému privilégiu.

Pri kontrole privilégií systém najprv kontroluje, či má proces najnižšie potrebné privilégium a potom pokračuje v kontrole privilégií smerom nahor pre prítomnosť silnejších privilégií. Napríklad proces s privilégiom **PV_AU_** má automaticky privilégium **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE** a proces s privilégiom **PV_ROOT** má automaticky všetky privilégiá zobrazené nižšie okrem privilégií **PV_SU_**.

PV_PROC_
Ekvivalent kombinácie všetkých ďalších privilégií **PV_PROC_**

PV_PROC_PRIO
Umožňuje procesu/vláknu zmeniť prioritu, politiku a ďalšie parametre plánovania

PV_PROC_CORE
Umožňuje procesu vytvoriť výpis z jadra

PV_PROC_RAC
Umožňuje procesu vytvoriť viac procesov, ako je limit na užívateľa

PV_PROC_RSET
Umožňuje pripojiť sadu prostriedkov (**rset**) k procesu alebo k vláknu

PV_PROC_ENV
Umožňuje procesu nastaviť informácie o užívateľoch v štruktúre užívateľov

PV_PROC_CKPT
Umožňuje procesu vytvoriť kontrolný bod alebo reštartovať iný proces

PV_PROC_CRED
Umožňuje procesu nastaviť atribúty splnomocnení procesu

PV_PROC_SIG
Umožňuje procesu poslať signál nesúvisiacemu procesu

PV_PROC_PRIV
Umožňuje procesu modifikovať alebo zobraziť sady privilégií priradené procesu

PV_PROC_TIMER
Umožňuje procesu odovzdať a používať časovače s jemnou zrnitosťou

PV_PROC_RTCLK
Umožňuje procesu prístup k hodinám času CPU

PV_PROC_VARS
Umožňuje procesu opakovane získať a aktualizovať parametre laditeľné procesom

PV_PROC_PDMODE
Umožňuje procesu zmeniť režim REAL adresára s oddielmi

Privilégia jadra:

V systéme Dôveryhodný systém AIX sú dostupné nasledujúce privilégiá jadra. K dispozícii je syntéza a popis každého privilégia a jeho použitia. Niektoré privilégiá vytvárajú hierarchiu, v ktorej jedno privilégium môže udeľovať všetky práva súvisiace s iným privilégiom.

Pri kontrole privilégií systém najprv vykoná kontrolu, aby určil, či má proces najnižšie potrebné privilégium, a potom pokračuje smerom nahor v hierarchii, aby skontroloval prítomnosť výkonnejších privilégií. Napríklad proces s privilégiom **PV_AU_** má automaticky privilégium **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE** a proces s privilégiom **PV_ROOT** má automaticky všetky nižšie uvedené privilégiá, okrem **PV_SU_**.

PV_KER_
Ekvivalent k všetkým ostatným kombinovaným privilégiám **PV_KER_**.

PV_KER_ACCT
Umožňuje procesu vykonať obmedzené operácie súvisiace s podsystémom účtov.

PV_KER_DR
Umožňuje procesu vyvolať operácie dynamických rekonfigurácií.

PV_KER_TIME
Umožňuje procesu upraviť systémové hodiny a čas.

PV_KER_RAC
Umožňuje procesu využívať pre segmenty zdieľanej pamäte veľké (nestránkovateľné) stránky.

PV_KER_WLM
Umožňuje procesu inicializovať a upraviť konfiguráciu WLM.

PV_KER_EWLM
Umožňuje procesu inicializovať a dotazovať prostredie eWLM.

PV_KER_VARS
Umožňuje procesu skúmať, alebo nastaviť parametre ladenia runtime jadra.

PV_KER_REBOOT
Umožňuje procesu vypnúť systém.

PV_KER_RAS
Umožňuje procesu konfigurovať, alebo zapisovať záznamy RAS, protokolovanie chýb, sledovanie a funkcie výpisov z pamäte.

PV_KER_LVM
Umožňuje procesu konfigurovať podsystém LVM.

PV_KER_NFS
Umožňuje procesu konfigurovať podsystém NFS.

PV_KER_VMM
Umožňuje procesu upraviť parametre swap a ostatné laditeľné parametre VMM v jadre.

PV_KER_WPAR
Umožňuje procesu konfigurovať oddiel pracovného zaťaženia.

PV_KER_CONF
Umožňuje procesu vykonávať rozličné operácie konfigurácie systému.

PV_KER_EXTCONF
Umožňuje procesu vykonávať rozličné konfiguračné úlohy v rozšíreniach jadra.

PV_KER_IPC
Umožňuje procesu zvýšiť hodnotu vyrovnávacej pamäte frontu správ IPC a umožniť pripojenie systémových volaní **shmget** s rozsahmi.

PV_KER_IPC_R
Umožňuje procesu čítať front správ IPC, sadu semaforov, alebo segmenty zdieľanej pamäte.

PV_KER_IPC_W
Umožňuje procesu zapisovať do frontu správ IPC, sady semaforov, alebo segmentov zdieľanej pamäte.

PV_KER_IPC_O
Umožňuje procesu čítať a prepisovať vlastníctvo DAC na všetkých objektoch IPC.

PV_KER_SECCONFIG
Umožňuje procesu nastaviť bezpečnostné príznaky jadra.

PV_KER_PATCH
Umožňuje procesu opraviť rozšírenia jadra.

Privilégiá označení:

V systéme Dôveryhodný systém AIX sú dostupné nasledujúce privilégiá označení. K dispozícii je syntéza a popis každého privilégia a jeho použitia. Niektoré privilégiá vytvárajú hierarchiu, v ktorej jedno privilégium môže udeľovať všetky práva súvisiace s iným privilégiom.

Pri kontrole privilégií systém najprv vykoná kontrolu, aby určil, či má proces najnižšie potrebné privilégium, a potom pokračuje smerom nahor v hierarchii, aby skontroloval prítomnosť výkonnejších privilégií. Napríklad proces s privilégiom **PV_AU_** má automaticky privilégium **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE** a proces s privilégiom **PV_ROOT** má automaticky všetky nižšie uvedené privilégiá, okrem **PV_SU_**.

PV_LAB_
Ekvivalent k všetkým ostatným kombinovaným privilégiám označení (**PV_LAB_***).

PV_LAB_CL
Umožňuje procesu upraviť SCL subjektu, s ohľadom na autorizáciu procesu.

PV_LAB_CLTL
Umožňuje procesu upraviť TCL subjektu, s ohľadom na autorizáciu procesu.

PV_LAB_LEF
Umožňuje procesu čítať z databázy označení.

PV_LAB_SLDG
Umožňuje procesu znížiť SL s ohľadom na autorizáciu procesu.

PV_LAB_SLDG_STR
Umožňuje procesu, s ohľadom na autorizáciu procesu, znížiť SL paketu.

PV_LAB_SL_FILE
Umožňuje procesu, s ohľadom na autorizáciu procesu, zmeniť SL objektu.

PV_LAB_SL_PROC
Umožňuje procesu, s ohľadom na autorizáciu procesu, zmeniť SL subjektu.

PV_LAB_SL_SELF
Umožňuje procesu, s ohľadom na autorizáciu procesu, zmeniť svoje vlastné SL.

PV_LAB_SLUG
Umožňuje procesu zvýšiť SL s ohľadom na autorizáciu procesu.

PV_LAB_SLUG_STR
Umožňuje procesu, s ohľadom na autorizáciu procesu, zvýšiť SL paketu.

PV_LAB_TL
Umožňuje procesu upraviť TL subjektu a objektu.

Privilégiá MAC:

V systéme Dôveryhodný systém AIX sú dostupné nasledujúce privilégiá MAC. K dispozícii je syntéza a popis každého privilégia a jeho použitia. Niektoré privilégiá vytvárajú hierarchiu, v ktorej jedno privilégium môže udeľovať všetky práva súvisiace s iným privilégiom.

Pri kontrole privilégií systém najprv vykoná kontrolu, aby určil, či má proces najnižšie potrebné privilégium, a potom pokračuje smerom nahor v hierarchii, aby skontroloval prítomnosť výkonnejších privilégií. Napríklad proces s privilégiom **PV_AU_** má automaticky privilégium **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE** a proces s privilégiom **PV_ROOT** má automaticky všetky nižšie uvedené privilégiá, okrem **PV_SU_**.

PV_MAC_
Ekvivalent k všetkým ostatným kombinovaným privilégiám MAC (**PV_MAC_***).

PV_MAC_CL
Umožňuje procesu preklenúť obmedzenia povolenej citlivosti.

PV_MAC_R_PROC
Pri získavaní informácií o procese umožňuje procesu preklenúť obmedzenia čítania MAC, za predpokladu, že označenie cieľového procesora je v rámci povoleného rozsahu pôsobiaceho procesu.

PV_MAC_W_PROC
Pri odosielaní signálu procesu umožňuje procesu preklenúť obmedzenia zápisu MAC, za predpokladu, že označenie cieľového procesora je v rámci povoleného rozsahu pôsobiaceho procesu.

PV_MAC_R
Umožňuje procesu preklenúť obmedzenia čítania MAC.

PV_MAC_R_CL
Ak sa označenie objektu nachádza v rámci rozsahu povoleného pre proces, umožňuje procesu preklenúť obmedzenia čítania MAC.

PV_MAC_R_STR
Umožňuje procesu preklenúť obmedzenia čítania MAC pri čítaní správy od STREAMu, za predpokladu, že sa označenie správy nachádza v rámci povoleného rozsahu procesu.

PV_MAC_W
Umožňuje procesu preklenúť obmedzenia zápisu MAC.

PV_MAC_W_CL
Ak sa označenie objektu nachádza v rámci povoleného rozsahu procesu, umožňuje procesu preklenúť obmedzenia zápisu MAC.

PV_MAC_W_DN
Ak je označenie procesu vyššie, než označenie objektu a označenie objektu sa nachádza v rámci povoleného rozsahu procesu, umožňuje procesu preklenúť obmedzenia zápisu MAC.

PV_MAC_W_UP
Ak je označenie procesu nižšie, než označenie objektu a označenie objektu sa nacháza v rámci povoleného rozsahu procesu, umožňuje procesu preklenúť obmedzenia zápisu MAC.

PV_MAC_OVRRD
Preklenie obmedzenia MAC pre súbory označené ako súbory vyňaté MAC.

Privilégiá MIC:

V systéme Dôveryhodný systém AIX sú dostupné nasledujúce privilégiá MIC. K dispozícii je syntéza a popis každého privilégia a jeho použitia. Niektoré privilégiá vytvárajú hierarchiu, v ktorej jedno privilégium môže udeľovať všetky práva súvisiace s iným privilégiom.

Pri kontrole privilégií systém najprv vykoná kontrolu, aby určil, či má proces najnižšie potrebné privilégium, a potom pokračuje smerom nahor v hierarchii, aby skontroloval prítomnosť výkonnejších privilégií. Napríklad proces s privilégiom **PV_AU_** má automaticky privilégium **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE** a proces s privilégiom **PV_ROOT** má automaticky všetky nižšie uvedené privilégiá, okrem **PV_SU_**.

PV_MIC
Umožňuje procesu preklenúť obmedzenia integrity.

PV_MIC_CL
Umožňuje procesu preklenúť obmedzenia povoleného rozsahu integrity.

Sieťové privilégiá:

Na Dôveryhodný systém AIX sú k dispozícii tieto sieťové privilégiá. Je poskytnutý prehľad a popis každého privilégia a jeho použitie. Niektoré privilégiá vytvárajú hierarchiu, kde jedno privilégium môže udeliť všetky práva priradené inému privilégiu.

Pri kontrole privilégií systém najprv kontroluje, či má proces najnižšie potrebné privilégium a potom pokračuje v kontrole privilégií smerom nahor pre prítomnosť silnejších privilégií. Napríklad proces s privilégiom **PV_AU_** má automaticky privilégium **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE** a proces s privilégiom **PV_ROOT** má automaticky všetky privilégiá zobrazené nižšie okrem privilégií **PV_SU_**.

PV_NET_
Ekvivalent kombinácie všetkých sieťových privilégií (**PV_NET_***)

PV_NET_CNTL
Umožňuje procesu modifikovať sieťové tabuľky

PV_NET_PORT
Umožňuje procesu vytvoriť väzby na obmedzený port

PV_NET_RAWSOCK
Umožňuje procesu mať priamy prístup na sieťovú vrstvu

PV_NET_CONFIG
Umožňuje procesu konfigurovať parametre siete

Privilégiá superužívateľa:

Na Dôveryhodný systém AIX sú k dispozícii tieto privilégiá superužívateľa. Je poskytnutý prehľad a popis každého privilégia a jeho použitie. Niektoré privilégiá vytvárajú hierarchiu, kde jedno privilégium môže udeliť všetky práva priradené inému privilégiu.

Pri kontrole privilégií systém najprv kontroluje, či má proces najnižšie potrebné privilégium a potom pokračuje v kontrole privilégií smerom nahor pre prítomnosť silnejších privilégií. Napríklad proces s privilégiom **PV_AU_** má automaticky privilégium **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE** a proces s privilégiom **PV_ROOT** má automaticky všetky privilégiá zobrazené nižšie okrem privilégií **PV_SU_**.

PV_SU_
Ekvivalent kombinácie všetkých ďalších privilégií superužívateľa (**PV_SU_***)

PV_SU_ROOT
Udelí procesu ekvivalent všetkých privilégií priradených štandardnému superužívateľovi

PV_SU_EMUL
Udelí procesu ekvivalent všetkých privilégií priradených štandardnému superužívateľovi, keď UID procesu je 0

PV_SU_UID
Spôsobí, že systémové volanie **getuid** vráti hodnotu 0

Rôzne privilégiá:

V systéme Dôveryhodný systém AIX sú dostupné nasledujúce rozličné privilégiá. K dispozícii je syntéza a popis každého privilégia a jeho použitia. Niektoré privilégiá vytvárajú hierarchiu, v ktorej jedno privilégium môže udeľovať všetky práva súvisiace s iným privilégiom.

Pri kontrole privilégií systém najprv vykoná kontrolu, aby určil, či má proces najnižšie potrebné privilégium, a potom pokračuje smerom nahor v hierarchii, aby skontroloval prítomnosť výkonnejších privilégií. Napríklad proces s privilégiom **PV_AU_** má automaticky privilégium **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** a **PV_AU_WRITE** a proces s privilégiom **PV_ROOT** má automaticky všetky nižšie uvedené privilégiá, okrem **PV_SU_**.

PV_ROOT
Udelí procesu ekvivalent všetkých ostatných privilégií okrem **PV_SU_** (a privilégií nižších, než **PV_SU_**).

PV_TCB
Umožňuje procesu upraviť dôveryhodnej cesty knižnice jadra.

PV_TP
Naznačuje, že proces je proces dôveryhodnej cesty a umožňuje akcie, ktoré sú limitované na procesy dôveryhodnej cesty.

PV_TP_SET
Umožňuje procesu nastaviť, alebo vymazať príznak dôveryhodnej cesty jadra.

PV_WPAR_CKPT
Umožňuje procesu vykonať kontrolný bod a reštartovať operácie oddielov pracovného zaťaženia.

PV_DEV_CONFIG
Umožňuje procesu konfigurovať rozšírenia a zariadenia systémového jadra.

PV_DEV_LOAD
Umožňuje procesu zaviesť a uvoľniť v systéme rozšírenia a zariadenia systémového jadra.

PV_DEV_QUERY
Umožňuje procesu dotazovať moduly jadra.

Riešenie problémov s Trusted AIX

Odpovede na bežné otázky, ktoré vám môžu pomôcť pri riešení problémov s Dôveryhodný systém AIX.

Ako sa môžem prihlásiť do Dôveryhodný systém AIX?
Dôveryhodný systém AIX vytvorí počas inštalácie troch administračných užívateľov s príslušnými rolami, ako je uvedené nižšie.

Heslá pre tieto kontá musíte nastaviť pri prvom zavedení systému po inštalácii Dôveryhodný systém AIX. Ak ste systém nainštalovali zo siete v režime bez výziev, heslá k týmto predvoleným kontám nájdete v nasledujúcej tabuľke.

Užívateľ	Heslo
isso	isso
sa	sa
so	so

Ako nastaviť atribút su na koreň?
Počas inštalácie Dôveryhodný systém AIX sa atribút **su** konta root nastaví na hodnotu **false**, aby žiadni užívateľ nemohol použiť toto konto. Aby bol prístup do tohto konta možný, isso a sa budú musieť zmeniť tento atribút koreňového konta na **true** pomocou príkazu **chuser**.

Ak je atribút **su** povolený pre koreň a heslo pre koreňové konto nie je nastavené, do koreňového konta bude mať prístup každý užívateľ v systéme. Ak tomu chcete predísť, odporúča sa nastaviť heslo koreňového konta pred resetovaním atribútu **su**.

Mám vytvoriť vlastných alebo použiť predvolených administrátorov?
Predvolení administrátori slúžia len na nastavenie systému pre účely prispôsobenia. Nie je to povinné, ale odporúča sa, aby boli tieto kontá použité len pre prispôsobenie systému.

Vytvorte si svojich vlastných troch administrátorov s príslušnými rolami isso, sa a so a vymažte alebo vypnite predvolených.

Prečo sa nemôžem prihlásiť do systému?
Ak sa pokúsite prihlásiť ako koreň (konto s uid 0) alebo ako iné konto s uid menším než 128, prístup bude zamietnutý. Tieto kontá sa nazývajú aj systémové kontá. Ak chcete mať prístup na systémové kontá, musíte sa na konto prihlásiť ako užívateľ nesystémového konta a **su**.

Zobrazí sa pri prihlasovaní chyba týkajúca sa súboru kódovania návestí?
Ak je súbor kódovania návestí poškodený, budete musieť zadať režim jedného užívateľa ako koreňový užívateľ. Koreňové konto je prístupné len v režime jedného užívateľa.

Pomocou príkazu **labck** skontrolujte, či je súbor kódovania návestí (/etc/security/enc/LabelEncodings) správny. Ak je súbor nesprávny, modifikujte ho a pred ukončením režimu jedného užívateľa ho znova skontrolujte pomocou príkazu **labck**.

Na overenie platnosti stavu systému spustite príkaz **trustchk** v interaktívnom režime (**trustchk -t ALL**).

Prečo nemôžem skompilovať žiadny program v systéme Dôveryhodný systém AIX používajúci rozhrania API knižníc Dôveryhodný systém AIX?
Vývojová sada nástrojov nie je štandardne nainštalovaná. Sadu súborov bos.mls.adt budete musieť nainštalovať z inštalačného média.

Ako odstránim zmeny vykonané v privilégiách príkazov, ktoré spôsobili, že tieto príkazy prestali správne fungovať?

Spustením príkazu **trustchk** v interaktívnom režime (**trustchk -t**) opravte privilégiá pre tieto príkazy.

Prečo nemám prístup do adresára /etc/security/enc?

Prostredie shell vyžaduje na prístup do adresára /etc/security/enc privilégiá PV_LAB_LEF a PV_MAC_R. Prideľte svojmu prostrediu shell uvedené privilégiá.

Ako vypnem príkaz trustchk pri zavedení?

Odstráňte alebo pripojte komentár do riadka príkazu trustchk v skripte /etc/rc.mls.

Ako dosiahnuť, aby systém nevyžadoval pri každom zavedení jeho autentifikáciu?

Možno ste zapli autentifikáciu zavedenia pre svoj systém. Môžete ju zakázať v podponuke Dôveryhodný systém AIX nástroja SMIT.

Prečo moja zmena nefunguje, keď chcem zmeniť SL objektu súborového systému?

Existuje niekoľko možností:

Vrátili /usr/sbin/settxattr nejaké chybové hlásenia?

Ak áno, vyhľadajte v nich bližšie informácie, napríklad:

Mali ste oprávnenie na spustenie /usr/sbin/settxattr?

Ak nie, skontrolujte si svoje privilégiá a oprávnenia.

Bola syntax správna?

Syntax nájdete na manuálovej stránke príkazu **settxattr**.

Existuje požadované návestie SL alebo jeho skratka?

Požadovanie "con a b" bude fungovať v systéme s predvoleným súborom kódovania návestí (/etc/security/enc/LabelEncodings), ale požadovanie "conf a b" nebude fungovať, aj keď oboje vyzerá ako logické skratky pre "confidential compartment A compartment B."

Museli ste pre návestie zložené z viacerých slov použiť úvodzovky?

settxattr -f sl=con <*názovsúboru*> a settxattr -f -a sl="con a b" <*názovsúboru*> budú fungovať, ale settxattr -a sl=con a b <*názovsúboru*> nebude.

Vrátil príkaz settxattr nejaké chybové hlásenia?

Ak nie, objekt súborového systému môže byť symbolickým odkazom. Ak je objekt, ktorý sa snažíte zmeniť, symbolickým odkazom, najprv určite, či chcete zmeniť SL samotného odkazu alebo objektu, na ktorý odkaz ukazuje. Príkaz **settxattr** nesleduje odkazy, ale namiesto toho nastavuje návestia samotného odkazu.

Ako mám nainštalovať aplikáciu tretej strany, aby v systéme riadne fungovala?

Ak ste nainštalovali aplikáciu tretej strany a táto riadne nefunguje, prístup do určitých vyhradených súborov alebo adresárov môže vyžadovať ďalšie privilégiá. Po vyhodnotení potreby aplikácie vstupovať do týchto vyhradených objektov stanovte potrebné privilégiá.

- Prideľte svojmu prostrediu shell PV_ROOT
- Spustite príkaz tracepriv -f -e <*príkaz tretej strany*>

Vypíše sa privilégium vyžadované aplikáciou. Pridajte ho do databázy privilegovaných príkazov pomocou príkazu **setsecattr**.

Prečo nemôžem spustiť niektoré príkazy?

Keďže väčšina príkazov je chránená oprávneniami, spustenie niektorých privilegovaných príkazov bude povolené len vtedy, ak má vyvolávajúci užívateľ zodpovedajúce oprávnenie. Toto môžete skontrolovať, ak zistíte, či v jednej z rolí aktivovaných pre aktuálnu reláciu existuje oprávnenie vyžadované na spustenie príkazu.

Skontrolujte svoje aktívne oprávnenia pomocou rolelist -ae a oprávnenia vyžadované príkazom pomocou lssecattr -c <*príkaz*>.

Prečo niektoré príkazy nezobrazujú návestia správne?
Väčšina týchto príkazov pri konverzii návestí do ľahko čitateľného tvaru a naopak dôveruje súboru /etc/security/enc/LabelEncodings. Ak bol tento súbor poškodený alebo modifikovaný, príkazy nemusia fungovať tak, ako by mali.

Bezpečnostné príznaky súboru

Bezpečnostné príznaky súboru majú vplyv na spôsob, ktorým sa na súbory pristupuje. Tieto príznaky sú uložené ako súčasť rozšírených atribútov (EA) samotného súboru. Bezpečnostné príznaky súboru sú definované v súbore hlavičiek.

FSF_APPEND
V prevádzkovom režime môžete súbor len pripojiť, ale nemôže ho zmeniť.

FSF_AUDIT
Súbor je označený ako súčasť auditovacieho podsystému. Ak chcete tieto súbory načítať alebo zapísať, proces musí mať privilégiá **PV_AU_READ** alebo **PV_AU_WRITE**.

FSF_MAC_EXMPT
EPS s privilégiom **PV_MAC_OVRRD** ignoruje obmedzenia MAC pri pokuse o prístup k objektu.

FSF_PDIR
Adresár je adresárom s oddielmi.

FSF_PSDIR
Adresár je podadresárom s oddielmi.

FSF_PSSDIR
Adresár je podadresárom podadresára s oddielmi.

FSF_TLIB
Objekt je označený ako súčasť dôveryhodnej knižnice. Počítač musí byť spustený v konfiguračnom režime alebo bezpečnostný príznak jadra **trustedlib_enabled** musí mať hodnotu OFF.

FSF_TLIB_PROC
Procesy označené ako procesy TLIB sa môžu pripájať len ku knižniciam oto ***.so**, ktoré majú nastavený príznak **TLIB**. Systém musí byť spustený v konfiguračnom režime alebo bezpečnostný príznak jadra **trustedlib_enabled** musí mať hodnotu OFF.

Príkazy Dôveryhodný systém AIX

Na riadenie systému Dôveryhodný systém AIX sa poskytujú bezpečnostné príkazy:

labck Skontroluje súbor LabelEncodings

getsecconf
Zobrazí bezpečnostné príznaky jadra

setsecconf
Zmení bezpečnostné príznaky jadra Dôveryhodný systém AIX

getsyslab
Zobrazí maximálne a minimálne návestia jadra

setsyslab
Nastaví maximálne a minimálne návestia jadra

getrunmode
Zobrazí aktuálny prevádzkový režim systému

setrunmode
Prepne prevádzkový režim systému

pdlink Pripojí súbory do všetkých podadresárov s oddielmi

pdmkdir
Vytvorí adresáre a podadresáre s oddielmi

pdmode
Vráti režim prístupu do aktuálneho adresára s oddielmi alebo spustí príkaz so zadaným režimom prístupu do adresára s oddielmi

pdrmdir
Odstráni adresáre s oddielmi a príslušné podadresáre

pdset Nastaví/zruší nastavenie (pod)adresárov s oddielmi

bootauth
Skontroluje, či oprávnený užívateľ bootuje systém

chuser Zmení atribúty vymazania užívateľa

lsuser Zobrazí atribúty vymazania užívateľa

chsec Zmení atribúty vymazania užívateľa a návestí portu

lssec Zobrazí atribúty vymazania užívateľa a návestí portu

trustchk
Skontroluje atribúty súborov

lstxattr
Zobrazí atribúty návestia a bezpečnostného príznaku súborov, procesov a objektov IPC

settxattr
Zmení atribúty návestia a bezpečnostného príznaku súborov, procesov a objektov IPC

Vyhlásenia

Tieto informácie boli vyvinuté pre produkty a služby ponúkané v USA.

Produkty, služby alebo súčasti popisované v tomto dokumente nemusí spoločnosť IBM ponúkať v iných krajinách. Informácie o produktoch a službách, ktoré sú k dispozícii vo vašej krajine, vám poskytne miestny zástupca spoločnosti IBM. Žiadny odkaz na produkt, program alebo službu spoločnosti IBM neznamená ani nenaznačuje, že je možné použiť len tento produkt, program alebo službu spoločnosti IBM. Namiesto nich sa môže použiť ľubovoľný funkčne rovnocenný produkt, program alebo služba, ktoré neporušujú žiadne právo duševného vlastníctva spoločnosti IBM. Užívateľ však zodpovedá za to, aby zhodnotil a overil používanie takéhoto produktu, programu alebo služby.

Spoločnosť IBM môže mať patenty alebo prihlášky patentov čakajúce na spracovanie, ktoré sa týkajú predmetu tohto dokumentu. Poskytnutie tohto dokumentu vám neudeľuje licenciu na tieto patenty. Otázky, týkajúce sa licencií, môžete zaslať písomne na:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
USA

V prípade otázok týkajúcich sa licencie na dvojbajtové informácie (DBCS) kontaktujte oddelenie intelektuálneho vlastníctva spoločnosti IBM Intellectual Property Department vo vašej krajine alebo ich pošlite písomne na adresu:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

SPOLOČNOSŤ INTERNATIONAL BUSINESS MACHINES POSKYTUJE TÚTO PUBLIKÁCIU „TAK, AKO JE“, BEZ AKÝCHKOĽVEK VÝSLOVNÝCH ALEBO MLČKY PREDPOKLADANÝCH ZÁRUK VRÁTANE, ALE BEZ OBMEDZENIA NA, PREDPOKLADANÝCH ZÁRUK NEPORUŠENIA PRÁV, PREDAJNOSTI ALEBO VHODNOSTI NA KONKRÉTNY ÚČEL. Niektoré štáty nepovoľujú zrieknutie sa výslovných ani mlčky predpokladaných záruk pri konkrétnych transakciách, preto sa na vás toto vyhlásenie nemusí vzťahovať.

Tieto informácie môžu obsahovať technické nepresnosti alebo typografické chyby. Tu uvádzané informácie sa periodicky menia; tieto zmeny budú začleňované do nových vydaní publikácie. Spoločnosť IBM môže kedykoľvek a bez prechádzajúceho upozornenia vykonať vylepšenia alebo zmeny v produktoch alebo programoch popísaných v tejto publikácii.

Akékoľvek odkazy na iné webové stránky než stránky spoločnosti IBM sa v tejto publikácii poskytujú len pre vaše pohodlie a v žiadnom prípade sa nemôžu chápať ako prejav súhlasu s obsahom týchto webových stránok. Materiály na týchto webových stránkach nie sú súčasťou materiálov k tomuto produktu spoločnosti IBM a ich použitie je výhradne na vaše vlastné riziko.

Spoločnosť IBM môže ľubovoľné vami poskytnuté informácie použiť alebo rozširovať spôsobom, ktorý uzná za vhodný, bez toho, aby jej tým vznikli akékoľvek záväzky voči vám.

Držitelia licencií tohto programu, ktorí si želajú mať informácie o tomto programe kvôli povoleniu: (i) výmeny informácií medzi nezávisle vytvorenými programami a inými programami (vrátane tohto programu) a (ii) spoločného používania vymenených informácií by mali kontaktovať:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
USA

Za primeraných podmienok, v niektorých prípadoch i za poplatok, môže spoločnosť IBM takéto informácie poskytnúť.

Spoločnosť IBM poskytuje licenčný program popísaný v tomto dokumente a všetky dostupné súvisiace licenčné materiály na základe podmienok zmluvy IBM Customer Agreement, zmluvy IBM International Program License Agreement alebo akejkoľvek rovnocennej zmluvy uzatvorenej medzi nami.

Údaje o výkone a príklady použitia sú uvádzané len na ilustratívne účely. Skutočný výkon sa môže líšiť v závislosti od konkrétnej konfigurácie a prevádzkových podmienok.

Informácie týkajúce sa produktov, ktoré nedodáva spoločnosť IBM, boli získané od dodávateľov týchto produktov, z nimi publikovaných textov alebo iných voľne dostupných zdrojov. Spoločnosť IBM tieto produkty netestovala a nemôže potvrdiť presnosť vyhlásení týkajúcich sa ich výkonu, kompatibility ani iných vyhlásení súvisiacich s produktmi nepochádzajúcimi od spoločnosti IBM. Otázky týkajúce sa schopností produktov, ktoré nie sú vlastnými produktmi spoločnosti IBM, je treba adresovať dodávateľom týchto produktov.

Vyhlásenia týkajúce sa budúceho smerovania alebo zámerov spoločnosti IBM môžu byť bez upozornenia zmenené alebo zrušené a predstavujú len ciele a zámery.

Všetky uvedené ceny stanovené spoločnosťou IBM predstavujú odporúčané maloobchodné ceny IBM, sú aktuálne a môžu sa zmeniť bez predchádzajúceho upozornenia. Ceny u jednotlivých predajcov sa môžu líšiť.

Tieto informácie sú určené len na účely plánovania. Tu uvedené informácie sa môžu zmeniť pred uvedením popisovaných produktov.

Tieto informácie obsahujú príklady údajov a správ, aké sa používajú v bežnej podnikovej praxi. Za účelom čo najväčšej zrozumiteľnosti tieto príklady obsahujú mená osôb, názvy spoločností, pobočiek a produktov. Všetky tieto mená a názvy sú fiktívne a akákoľvek ich podobnosť s menami osôb a názvami spoločností je úplne náhodná.

LICENCIA NA AUTORSKÉ PRÁVA:

Tieto informácie obsahujú vzorové aplikačné programy v zdrojovom jazyku, ktoré demonštrujú programovacie techniky v rozličných operačných platformách. Tieto vzorové programy môžete kopírovať, upravovať a distribuovať za účelom vývoja, používania, podpory predaja alebo distribuovania aplikačných programov vyhovujúcich aplikačnému programovému rozhraniu operačnej platformy, pre ktorú boli tieto vzorové programy napísané, a to v akejkoľvek forme a bez toho, že by vám tým vznikol finančný záväzok voči spoločnosti IBM. Tieto vzorové programy neboli dôkladne testované za všetkých podmienok. Spoločnosť IBM preto nemôže zaručiť ani predpokladať spoľahlivosť, prevádzkyschopnosť alebo funkčnosť týchto programov. Tieto vzorové programy sa poskytujú „TAK, AKO SÚ“ bez záruky akéhokoľvek druhu. Spoločnosť IBM nebude niesť zodpovednosť za žiadne škody, ktoré vzniknú v dôsledku používania týchto vzorových programov.

Všetky kópie alebo ľubovoľné časti týchto vzorových programov, ako aj všetky odvodené diela musia obsahovať nasledujúce vyhlásenie o autorských právach:

© (názov vašej spoločnosti) (rok).

Časti tohto kódu sú odvodené zo Vzorových programov spoločnosti IBM Corp.

© Copyright IBM Corp. _uveďte rok alebo roky_.

Ochrana osobných údajov

V softvérových produktoch IBM vrátane riešení SaaS (Software as a Service) („Ponuky softvéru“) sa môžu používať objekty cookie a iné technológie s cieľom zhromažďovať informácie o používaní produktu, zlepšiť skúsenosti koncových užívateľov, prispôsobiť komunikáciu s koncovými užívateľmi a iné účely. Vo väčšine prípadov tieto Ponuky softvéru nezhromažďujú žiadne informácie umožňujúce identifikáciu osôb. Niektoré Ponuky softvéru vám môžu pomôcť pri zhromažďovaní informácií umožňujúcich identifikáciu osôb. Ak táto Ponuka softvéru používa objekty cookie s cieľom zhromažďovať informácie umožňujúce identifikáciu osôb, nižšie nájdete podrobné informácie o tom, ako táto ponuka používa objekty cookie.

Táto Ponuka softvéru nepoužíva objekty cookie alebo iné technológie s cieľom zhromažďovať informácie umožňujúce identifikáciu osôb.

Ak nasadené konfigurácie tejto Ponuky softvéru umožňujú vám ako zákazníkovi zhromažďovať informácie umožňujúce identifikáciu osôb od koncových užívateľov prostredníctvom objektov cookie alebo iných technológií, mali by ste požiadať o právnu pomoc v súvislosti s právnymi predpismi, ktoré sa vzťahujú na takéto zhromažďovanie údajov vrátane požiadaviek týkajúcich sa upozornenia na toto zhromažďovanie informácií a súhlasu s ním.

Bližšie informácie o používaní rozličných technológií vrátane objektov cookie na tieto účely nájdete v Zásadách ochrany osobných údajov spoločnosti IBM na adrese http://www.ibm.com/privacy a Prehlásení o online ochrane osobných údajov IBM na adrese http://www.ibm.com/privacy/details v časti s názvom „Objekty cookie, Web Beacon a iné technológie“ a v dokumente „Vyhlásenie o ochrane osobných údajov v softvérových produktoch a ponukách SaaS spoločnosti IBM“ na adrese http://www.ibm.com/software/info/product-privacy.

Ochranné známky

IBM, logo IBM a ibm.com sú ochranné známky alebo registrované ochranné známky spoločnosti International Business Machines Corp., zaregistrované v rôznych jurisdikciách na celom svete. Ďalšie názvy produktov a služieb môžu byť ochranné známky IBM alebo iných spoločností. Aktuálny zoznam ochranných známok spoločnosti IBM nájdete na stránke Informácie o autorských právach a ochranných známkach na adrese www.ibm.com/legal/copytrade.shtml.

Linux je registrovaná ochranná známka Linusa Torvaldsa v USA alebo iných krajinách.

Microsoft a Windows sú ochranné známky spoločnosti Microsoft Corporation v USA alebo v iných krajinách.

Java a všetky s ňou súvisiace ochranné známky a logá sú ochranné známky alebo registrované ochranné známky spoločnosti Oracle alebo jej pridružených spoločností.

UNIX je registrovaná ochranná známka spoločnosti Open Group v USA alebo iných krajinách.

Index

Špeciálne znaky

A

B

C

D

E

F

H

I

K

L

M

N

O

P

R

S

Š

T

U

V

X

Z

Vytlačené v USA