AIX версии 7.2

Защита

AIX версии 7.2

Защита

Примечание

Перед началом работы с этим изданием и описанным в нем продуктом ознакомьтесь с информацией, приведенной в разделе “Примечания” на стр. 515.

Данное издание относится к AIX версии 7.2, а также ко всем последующим выпускам и модификациям, если в соответствующих изданиях не будет оговорено обратное.

Содержание

Об этом документе

В этих разделах, адресованных системным администраторам, приведена информация о защите файлов, систем и сетей. Они включает в себя инструкции по выполнению таких задач, как укрепление безопасности системы, изменение прав доступа, настройка способов идентификации и настройка общих критериев оценки защиты. Данные разделы можно найти и на компакт-диске документации, который поставляется вместе с операционной системой.

Выделение текста

В данном документе применяются следующие специальные обозначения:

Полужирный шрифт	Этим шрифтом выделены команды, функции, ключевые слова, файлы, структуры, каталоги и другие элементы, имена которых предопределены в системе. Кроме того, этим шрифтом выделены графические объекты, выбираемые пользователем: кнопки, метки и значки.
Курсив	Этим шрифтом выделены параметры, фактические имена или значения которых указываются пользователем.
`Непропорциональный шрифт`	Этим шрифтом выделены примеры конкретных значений, образцы фрагментов текста, которые могут быть показаны на экране, примеры программного кода, схожие с реальными, системные сообщения и информация, вводимая пользователем.

Учет регистра символов в AIX

В операционной системе AIX учитывается регистр символов, т.е. различаются прописные и строчные буквы. Например, команда **ls** выдает список файлов. При вводе команды `LS` отобразится сообщение `Команда не найдена`. Аналогично, имена файлов **FILEA**, **FiLea** и **filea** считаются разными, даже если эти файлы расположены в одном каталоге. Во избежание нежелательных последствий всегда контролируйте регистр вводимых символов.

ISO 9000

При разработке и производстве данного продукта использовались зарегистрированные системы ISO 9000.

Защита

Операционная система AIX включает в себя инструкции по выполнению таких задач, как повышение безопасности системы, изменение прав доступа, настройка способов идентификации и настройка Общих критериев оценки защиты. Данные разделы можно найти и на компакт-диске документации, который поставляется вместе с операционной системой.

Информация, связанная с данной:

- Computer Emergency Response Team в Университете Карнеги-Мелон (CERT)
- Forum of Incident Response and Security Teams (FIRST)
- Center for Education and Research in Information Assurance and Security (CERIAS)

Новое в книге Защита

Новая и значительно измененная информация в книге Защита.

Обозначение дополнений и изменений

В данном файле PDF новая и измененная информация может выделяться значками (|) в левом поле.

Январь 2017 года

Ниже приведено краткое описание изменений, внесенных в разделы из этой книги:

- Добавлена информация о событиях контроля в раздел “События контроля” на стр. 143.
- Добавлена информация об образах OpenSSH в раздел “Образы OpenSSH” на стр. 206.

Защита базовой операционной системы

В этом разделе приведена информация о защите системы без учета подключения к сети.

В этой части описана установка системы с включенными параметрами защиты, а также рассказано о защите AIX от несанкционированного доступа пользователей к системе.

Установка и настройка защищенной системы

Установка и настройка AIX включает в себя ряд процедур.

Защищенная компьютерная база

Системный администратор должен определить степень доверия к каждой программе. При этом следует обязательно учитывать значимость информационных ресурсов в системе.

Защищенная компьютерная база (TCB) - это часть системы, отвечающая за выполнение стратегии защиты информации. TCB позволяет предоставлять пользователям защищенный доступ только к определенным каталогам и файлам. Функции TCB можно включить только при установке операционной системы. Для установки TCB в существующей системе необходимо выполнить установку с сохранением. TCB предоставляет доступ к защищенной оболочке, защищенным процессам и защищенной клавише внимания (SAK).

Проверка TCB:

При недостаточной защите файлов защищенной компьютерной базы (TCB) или неправильных значениях в файлах конфигурации уровень защиты системы заметно снижается.

Команда **tcbck** проверяет состояние защищенной компьютерной базы. Команда **tcbck** отслеживает соответствующую информацию, читая файл `/etc/security/sysck.cfg`. В это файле описаны все файлы TCB, файлы конфигурации и защищенные команды.

Файл `/etc/security/sysck.cfg` находится в системе и, следовательно, может быть изменен хакерами. Обязательно создавайте автономную копию этого файла, предназначенную только для чтения, после каждого изменения исходного файла. Перед каждой проверкой копируйте этот файл с архивного носителя на диск.

Структура файла sysck.cfg:

Команда **tcbck** считывает из файла `/etc/security/sysck.cfg` список файлов, подлежащих проверке. Каждой защищенной программе в системе соответствует раздел файла `/etc/security/sysck.cfg`.

У каждого раздела есть следующие атрибуты:

Атрибут	Описание
acl	Текстовая строка, представляющая список управления доступом к файлу. Формат этого списка должен соответствовать формату вывода команды **aclget**. При несовпадении списков прав доступа команда **sysck** присваивает соответствующему атрибуту файла указанное здесь значение с помощью команды **aclput**. **Примечание:** Атрибуты SUID, SGID и SVTX должны соответствовать значениям, указанным для атрибута mode, если они указаны.
class	Имя группы файлов. Этот атрибут позволяет проверить все файлы из класса, указав в команде **tcbck** только имя класса. Можно указать несколько имен классов через запятую.
group	ИД группы или имя группы файла. При несовпадении имен групп файлов команда **tcbck** присваивает параметру группы файла указанное здесь значение.
links	Список связей в этим файлом, разделенных запятыми. Если какой-либо из указанных путей не связан с данным файлом, то команда **tcbck** восстанавливает связь. Если параметр *tree* не указан, то команда **tcbck** выдает сообщения о наличии других связей, но не определяет их имена. Если указан параметр *tree*, то команда **tcbck** также показывает все дополнительные пути, связанные с этим файлом.
mode	Список значений, разделенных запятыми. Допустимы значения SUID, SGID, SVTX и TCB. Права доступа к файлу должны указываться последними и могут быть представлены либо в виде октета, либо в виде 9-символьной строки. Например, допустимы значения 755 или rwxr-xr-x. При несовпадении прав доступа команда **tcbck** присваивает соответствующему атрибуту файла указанное здесь значение.
owner	ИД пользователя или имя владельца файла. При несовпадении имен владельцев файлов команда **tcbck** присваивает параметру владельца файла указанное здесь значение.
program	Список значений, разделенных запятыми. Первое значение - это имя пути к проверяющей программе. Остальные значения передаются этой программе в качестве аргументов. **Примечание:** Первый аргумент всегда будет равен *-y*, *-n*, *-p* или *-t*, в зависимости от флага, с которым была вызвана команда **tcbck**.
source	Имя файла, из которого был скопирован данный файл перед его проверкой. Если для обычного файла, каталога или поименованного конвейера указано пустое значение, то создается пустая версия файла, если она еще не существует. Для файлов устройств создается новый особый файл с тем же типом устройства.
symlinks	Список символических ссылок на этот файл, разделенных запятыми. Если какой-либо из указанных путей не является символической ссылкой на файл, то эта ссылка создается командой **tcbck**. Если задан параметр *tree*, то команда **tcbck** также показывает все дополнительные пути, являющиеся символическими ссылками на этот файл.

Если некоторый атрибут раздела файла `/etc/security/sysck.cfg` не указан, то соответствующая проверка не выполняется.

Применение команды tcbck:

Команда **tcbck** служит для проверки правильности установки файла, подлежащего защите; для проверки отсутствия в файловой системе файлов, нарушающих ее защиту, а также для обновления, добавления или удаления защищенных файлов.

Как правило, команда **tcbck** применяется для выполнения следующих действий:

- Проверка правильности установки файлов, связанных с защитой
- Поиск в системе файлов, явно нарушающих защиту
- Обновление, добавление или удаление защищенных файлов

Действие команды **tcbck** зависит от способа использования:

- Обычный вызов
  - Неинтерактивный вызов при инициализации системы
  - Вызов с помощью команды **cron**
- Интерактивное использование
  - Проверка конкретных файлов и классов файлов
- Максимальное использование
  - Сохранение файла `sysck.cfg` на съемном носителе и периодическое восстановление для проверки системы.

Хотя шифрование и не применяется, TCB использует команду **sum** для проверки контрольной суммы. Контрольной суммой базы данных TCB можно управлять вручную, с помощью другой команды, например **md5sum**. Эта команда поставляется в пакете textutilsАдминистратор пакетов RPM на компакт-диске *AIX Toolbox for Linux Applications*.

Проверка защищенных файлов:

Для проверки и исправления всех файлов базы данных tcbck и создания протокола ошибок служит команда **tcbck**.

Для проверки и исправления всех файлов в базе данных tcbck с сообщением обо всех ошибках введите:

```
tcbck -y ALL
```

Команда **tcbck** будет проверять те файлы базы данных tcbck, которые указаны в файле `/etc/security/sysck.cfg`.

Для того чтобы это действие выполнялось автоматически при инициализации системы, внесите эту команду в команду **/etc/rc**.

Проверка дерева файловой системы:

При любом подозрении на нарушение целостности файловой системы проверяйте ее с помощью команды **tcbck**.

Для проверки дерева файловой системы выполните следующие действия:

```
tcbck -t tree
```

Если команда **tcbck** применяется со значением *tree*, то будут проверены все файлы в файловой системе (на это может потребоваться много времени). При обнаружении командой **tcbck** потенциального нарушения защиты, вы можете изменить подозрительные атрибуты файла. Кроме того, для всех остальных файлов в системе выполняются следующие процедуры проверки:

- Если владельцем файла является root и для файла задан атрибут SetUID, то этот атрибут очищается.

- Если исполняемый файл входит в административную группу и для файла задан атрибут SetGID, то этот атрибут очищается.
- Если для файла задан атрибут **tcb**, то он очищается.
- Если это файл устройства (символьный или блочный), то он удаляется.
- Если файл является дополнительной связью с путем, указанным в файле /etc/security/sysck.cfg, то эта связь удаляется.
- Если файл является дополнительной символической ссылкой на путь, указанный в файле /etc/security/sysck.cfg, то эта ссылка удаляется.

  Примечание: Перед вызовом команды **tcbck** для всех устройств должны быть созданы записи в файле /etc/security/sysck.cfg. В противном случае система будет повреждена. Для добавления защищенных устройств в файл /etc/security/sysck.cfg служит флаг **-l**.

Внимание: *Не вызывайте* команду **tcbck -y tree**. Эта опция удаляет и отключает все устройства, не перечисленные в TCB, что может привести систему в нерабочее состояние.

Добавление защищенной программы:

Добавить защищенную программу в файл /etc/security/sysck.cfg можно с помощью команды **tcbck**.

Для добавления защищенной программы в файл /etc/security/sysck.cfg введите:

```
tcbck -a Путь [Атрибут=Значение]
```

В командной строке необходимо указать только те атрибуты, которые не следуют из текущего состояния файла. Все имена атрибутов указаны в файле /etc/security/sysck.cfg.

Например, следующая команда регистрирует новую программу SetUID root с именем /usr/bin/setgroups, связанную с /usr/bin/getgroups:

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

Для добавления административных пользователей jfh и jsl, а также группы администраторов developers, которая будет проверяться при отслеживании защиты файла /usr/bin/abc, введите:

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

После установки программы вы можете не знать, какие новые файлы зарегистрированы в файле /etc/security/sysck.cfg. Для получения списка таких файлов введите следующую команду:

```
tcbck -t tree
```

Эта команда показывает список всех файлов, зарегистрированных в файле /etc/security/sysck.cfg.

Удаление защищенной программы:

После удаления из системы файла, описанного в /etc/security/sysck.cfg, удалите его описание из этого файла.

Например, если вы удалили программу /etc/cvid, то следующая команда вызовет сообщение об ошибке:

```
tcbck -t ALL
```

Будет показано следующее сообщение:

```
3001-020 Файл /etc/cvid не найден.
```

Описание программы все еще находится в файле /etc/security/sysck.cfg. Для его удаления введите следующую команду:

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

Настройка дополнительных параметров защиты:

В этом разделе описана настройка дополнительных параметров для защищенной компьютерной базы (TCB).

Ограничение доступа к терминалу:

В этом разделе описываются способы настройки ограничения доступа к терминалу в операционной системе.

Команды **getty** и **shell** изменяют владельца и режим терминала, чтобы запретить доступ к нему ненадежных программ. Операционная система позволяет настроить исключительный доступ к терминалу.

Работа с защищенной клавишей внимания:

Защищенное соединение устанавливается при нажатии зарезервированной клавиши SAK (Ctrl-X, затем Ctrl-R).

Примечание: SAK следует применять с особой осторожностью, поскольку она останавливает все процессы, пытающиеся обратиться к терминалу, а также все связи с ним (например, `/dev/console` может быть связана с `/dev/tty0`).

Это следует делать в следующих случаях:

- При входе в систему

  При нажатии SAK:
  - Если появляется новое окно входа в систему, то защищенное соединение установлено.
  - Если появляется приглашение защищенной оболочки, то начальное окно входа в систему было незащищенной программой, которая, возможно, пыталась украсть ваш пароль. Определите текущего пользователя терминала, для чего введите команду **who** и выйдите из системы.
- Если введенная команда должна запускать защищенную программу. Некоторые примеры:
  - Работа от имени пользователя root. Станьте пользователем root только после установления защищенного соединения. Это позволит гарантировать, что с правами пользователя root не будут запущены незащищенные программы.
  - Вызов команд **su**, **passwd** и **newgrp**. Вызывайте эти команды только после установления защищенного соединения.

Настройка защищенной клавиши внимания:

Настройка защищенной клавиши внимания нужна для создания защищенного соединения.

Каждый терминал можно независимо настроить таким образом, чтобы каждое нажатие SAK приводило к созданию защищенного соединения. Для этого в файле `/etc/security/login.cfg` необходимо указать атрибут **sak_enabled**. Если значение атрибута - True, то SAK включена.

Если для связи должен применяться конкретный порт (например, в команде **uucp**), то в файле `/etc/security/login.cfg` необходимо указать для порта следующее значение:

```
sak_enabled = false
```

Эта строка (или отсутствие записи в этом разделе) отключает SAK для терминала.

Для включения SAK терминала добавьте в соответствующий раздел следующую строку:

```
sak_enabled = true
```

Защищенное выполнение

Защищенное выполнение (TE) представляет собой набор функций, которые используются для проверки целостности системы и реализуют дополнительные стратегии защиты, что в совокупности может использоваться для повышения уровня защиты всей системы.

Обычно для повреждения системы злонамеренный пользователь получает доступ к ней и устанавливает `троянских коней`, наборы rootkit или подделывает важные файлы защиты. Система становится уязвимой, и он может ее использовать. Основной целью набора функций защищенного выполнения является предупреждение таких действий или, в худшем случае, возможность установить, произошел ли в системе подобный инцидент. Используя функции, предоставляемые защищенным выполнением, системный администратор может выбрать либо определенный набор исполняемых программ, выполнение которых разрешено, либо набор расширений ядра, которые имеют разрешение на загрузку. Также эти функции можно использовать для контроля состояния защиты системы и выявления измененных файлов, тем самым повышая уровень защиты системы затрудняя для злонамеренного пользователя нанесение вреда системе. Функции набора TE можно объединить в следующие группы:

- Управление базой данных надежных сигнатур
- Контроль целостности базы данных надежных сигнатур
- Настройка стратегий защиты
- Путь защищенного выполнения и защищенной библиотеки

Примечание: Функциональность TCB уже представлена в операционной системе AIX. TE является более мощным и совершенным механизмом, который частично совпадает с функциональностью TCB и предоставляет дополнительные стратегии защиты для усиления контроля целостности системы. В то время как Защищенная компьютерная база остается доступной, защищенное выполнение представляет собой новую и более развитую концепцию проверки и защиты целостности системы.

Управление базой данных надежных сигнатур:

Подобно Защищенной компьютерной базе (TCB), существует база данных, которая используется для хранения основных параметров защиты защищенных файлов системы. Эта база данных называется Базой данных надежных сигнатур (TSD) и хранится в файле `/etc/security/tsd/tsd.dat`.

Защищенным файлом является файл, который важен с точки зрения защиты системы и в случае несанкционированного доступа может поставить под угрозу безопасность системы в целом. Обычно этому определению соответствуют следующие файлы:

- Ядро (операционной системы)
- Все корневые программы setuid
- Все корневые программы setgid
- Любые программы, которые запускаются исключительно корневым пользователем или членом системной группы
- Любые программы, которые должны запускаться администратором по защищенному соединению (например, команда **ls**)
- Файлы конфигурации, управляющие работой системы
- Любые программы, которые запускаются с правами доступа на изменение ядра системы или системных файлов конфигурации

С каждым защищенным файлом должен быть связан файл настройки или определение файла, которые расположены в Базе данных надежных сигнатур (TSD). Файл может быть отмечен как защищенный путем добавления его определения в в TSD с помощью команды **trustchk**. Команду **trustchk** можно использовать для добавления, удаления или просмотра записей TSD.

База данных надежных сигнатур:

База данных надежных сигнатур является базой данных, которая используется для хранения основных параметров защиты защищенных файлов системы. Эта база данных расположена в каталоге `/etc/security/tsd/tsd.dat`.

С каждым защищенным файлом должен быть связан файл настройки или определение файла, которые расположены в Базе данных надежных сигнатур (TSD). Каждый защищенный файл связан с уникальным криптографическим хэшем и цифровой подписью. Криптографический хэш набора защищенных файлов по умолчанию создается с помощью алгоритма SHA-256, а цифровая подпись - с помощью RSA с использованием среды компоновкиAIX и упаковывается как часть наборов установочных файлов AIX. Эти значения хэш-функции и подписи поставляются как части соответствующих установочных образов AIX и хранятся в базе данных защищенных программ (`/etc/security/tsd/tsd.dat`) в целевой системе в файле настройки, который может иметь следующий формат:

```
/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 Владелец файла. Это значение вычисляется с помощью команды **trustchk**, при добавлении файла в TSD.

group Группа файла. Это значение вычисляется командой **trustchk**.

mode Список значений, разделенных запятыми. Допустимыми значениями являются **SUID** (установлен бит SUID), **SGID** (установлен бит SGID), **SVTX** (установлен бит SVTX) и **TCB** (Защищенная компьютерная база). Права доступа к файлу должны указываться последними и могут быть представлены в виде октета. Например, файл с режимом **uid** и битами доступа **rwxr-xr-x** режим будет иметь значение **SUID, 755**. Это значение вычисляется командой **trustchk**.

type Тип файла. Это значение вычисляется командой **trustchk**. Допустимы значения **FILE**, **DIRECTORY**, **MPX_DEV**, **CHAR_DEV**, **BLK_DEV**, и **FIFO**.

hardlinks
Список жестких ссылок на файл. Это значение не может быть вычислено командой **trustchk**. Оно должно быть указано пользователем при добавлении файла в базу данных.

symlinks
Список символьных ссылок на файл. Это значение не может быть вычислено командой **trustchk**. Оно должно быть указано пользователем при добавлении файла в базу данных.

size Определяет размер файла. Значение **VOLATILE** - файл изменяется часто.

cert_tag
Поле связывает цифровую подпись файла с сертификатом, которым будет проверяться эта подпись. В этом поле содержится идентификатор сертификата, который можно вычислить командой **trustchk** во время добавления файла в TSD. Сертификаты хранятся в каталоге `/etc/security/certificates`.

signature
Цифровая подпись файла. Значение **VOLATILE** указывает на то, что файл изменяется часто. Значение поля вычисляется командой **trustchk**.

hash_value
Криптографический хэш файла. Значение **VOLATILE** указывает на то, что файл изменяется часто. Значение поля вычисляется командой **trustchk**.

minslabel
Определяет метку минимальной чувствительности объекта.

maxslabel
Определяет метку максимальной чувствительности объекта (действует в системе Trusted AIX). Этот атрибут не применяется к обычным файлам и файлам типа fifo.

intlabel
Определяет метку целостности объекта (действует в системе Trusted AIX).

accessauths
Определяет права доступа к объекту (действует в системе Trusted AIX).

innateprivs
Определяет изначальные права доступа для файла.

proxyprivs
Определяет для файла права доступа через прокси-сервер.

authprivs
Определяет права доступа, назначаемые пользователю после санкционирования доступа.

secflags
Определяет флаги защиты, связанные с объектом.

t_accessauth
Определяет дополнительную Trusted AIX с особыми правами доступа на базе Многоуровневой защиты (MLS) (действует в системе Trusted AIX).

t_innateprivs
Задает дополнительный Trusted AIX с внутренними правами доступа к файлу в MLS (действителен только в системе Trusted AIX).

t_proxyprivs
Задает дополнительный Trusted AIX с правами доступа proxy к файлу в MLS (действителен только в системе Trusted AIX).

t_authprivs
Задает дополнительный Trusted AIX с правами доступа в MLS, которые присваиваются пользователю после заданных процедур идентификации (действителен только в системе Trusted AIX).

t_secflags
Задает дополнительный Trusted AIX с флагами защиты в MLS, связанными с объектом (действителен только в системе Trusted AIX).

Когда в TSD добавляется новая запись, то содержащиеся в доверенном файле символьные или жесткие ссылки в TSD можно добавить с помощью команд **symlinks**, **hardlinks** и **trustchk**. Если ожидается частое изменение добавляемого файла, то введите в командной строке ключевое слово VOLATILE. В этом случае

команда **trustchk** не будет вычислять значения полей**hash_value** и **signature** при создании определения файла для добавления в TSD. При проверке целостности этого файла поля **hash_value** и **signature** игнорируются.

При добавлении в TSD определений обычных файлов необходимо указать личный ключ (формат ASN.1/DER). Для этого укажите флаг **-s** с цифровым сертификатом и флаг **-v** с соответствующим открытым ключом. Личный ключ используется для создания сигнатуры файла, а впоследствии аннулируется. Пользователь должен самостоятельно обеспечить безопасное хранение этого ключа. Сертификат сохраняется в хранилище сертификатов в файле `/etc/security/certificates` для проверки сигнатур при запросе проверки целостности. Поскольку вычисление сигнатуры невозможно для файлов, которые не являются обычными, например файлов каталогов и устройств, при добавлении таких файлов в TSD не обязательно указывать личный ключ и сертификат.

С помощью опции **-f** в TSD можно добавить уже сформированное определение файла. Команда **trustchk** не будет ничего вычислять и сохранит определение в TSD без проверки. В такой ситуации пользователь лично ответственен за работоспособность определений файлов.

Поддержка проверки библиотеки

Для поддержки проверки библиотеки добавьте файл `tsd.dat` в каталог `/etc/security/tsd/lib/`. Имя базы данных хранится в `/etc/security/tsd/lib/lib.tsd.dat`. В базе данных хранятся исключительно те библиотеки, у которых соответствующие доверенные библиотеки хранят свои разделы в файлах `.o`. Раздел каждого файла `.o` библиотеки хранится в формате, как указано в следующем примере.

Если файл `strcmp.o`, относящийся к библиотеке `libc.a`, имеет тип `.o`, то раздел файла `strcmp.o` в файле `/etc/security/tsd/lib/lib.tsd.dat` имеет следующий формат:

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

В базе данных хранятся записи, соответствующие полям **type**, **hash size**, **cert tag** и **signature** файла `.o`. Хэш-код раздела, соответствующего библиотеке, обновляется в файле `/etc/security/tsd/tsd.dat`. Значения атрибутов генерируются динамически во время компоновки и переносятся в базу данных `/etc/security/tsd/lib/lib.tsd.dat` во время установки.

В файле `/etc/security/tsd/tsd.dat` отображаются изменения атрибута **type** раздела библиотеки, поскольку атрибуты `LIB`, **size** и **signature** пусты. В данный момент, если атрибуты **size**, **hash**, **signature** имеют значение **dynamica**, то они обрабатываются как **VOLATILE**. Таким образом, во время запуска системы проверка библиотек не выполняется. Начиная с выпуска AIX 6.1.0, разделы **size**, **hash** и **signature** доверенных библиотек рассчитываются на основании файлов `.o` соответствующей библиотеки. Во время установки, в базе данных `tsd.dat` устанавливаются рассчитанные значения, а соответствующие разделам файлы `.o` сохраняются в базе данных `/etc/security/tsd/lib/lib.tsd.dat`.

Удаленный доступ к базе данных TE:

Централизованные стратегии База данных надежной подписи (TSD) и Надежное выполнение (TE) могут быть реализованы в среде системы с помощью их сохранения в LDAP.

Базы данных, которые управляют стратегиями TSD и TE, хранятся независимо в каждой системе. Централизованные стратегии TSD и TE AIX хранятся в LDAP, поэтому они могут управляться централизованно. Использование централизованных стратегий TSD и TE позволяет обеспечить хранение главных копий стратегий в LDAP и возможность обновления клиентов при каждой переустановке, обновлении или нарушении защиты на клиентах. Централизованные стратегии TE позволяют обеспечивать стратегии TE из одного расположения без необходимости обновления каждого клиента по-отдельности. Централизованными стратегиями TSD намного легче управлять, чем нецентрализованными.

Утилиты AIX можно использовать для экспорта данных локальных стратегий TSD и TE в LDAP, настройки клиентов для использования данных стратегий TSD и TE в LDAP, управления поиском данных стратегий TSD и TE и управления данными LDAP из системы клиента. В следующих разделах более подробно описываются эти функции.

Экспорт данных стратегий TSD и TE на LDAP:

Для использования LDAP в качестве централизованного хранилища стратегий TSD и TE необходимо заполнить данные стратегии на сервере LDAP.

На сервере LDAP должны быть установлены схемы стратегий TSD и TE для LDAP, перед тем как клиенты LDAP смогут использовать сервер для данных стратегий. Схемы стратегий TSD и TE для LDAP доступны в системе AIX в файле **/etc/security/ldap/sec.ldif**. Схему сервера LDAP необходимо обновить с использованием этого файла при помощи команды **ldapmodify**.

Для того чтобы определить версию баз данных TE на сервере LDAP и информировать клиентов LDAP о ней, необходимо установить атрибут **databasename** в файле **/etc/nscontrol.conf**. Атрибут **databasename** принимает имя как значение, и он используется командой **tetoldif** при генерации формата ldif.

Выполните команду **tetoldif** для чтения данных из файлов стратегий TSD и TE и вывода стратегий в формате, применимом для LDAP. Сгенерированный командой **tetoldif** вывод можно сохранить в файле в формате ldif, а затем использовать для заполнения данных на сервере LDAP с помощью команды **ldapadd**. Для создания данных стратегий TSD и TE для LDAP команда **tetoldif** использует следующие базы данных:

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

Конфигурация клиента LDAP для стратегий TSD и TE:

Для использования данных стратегий TSD и TE на LDAP следует настроить систему в качестве клиента LDAP.

Для настройки системы как клиента LDAP выполните команду AIX **/usr/sbin/mksecldap**. Команда **mksecldap** производит динамический поиск по указанному серверу LDAP и определяет расположение данных стратегий TSD и TE, сохраняя результаты в файле **/etc/security/ldap/ldap.cfg**.

После успешной настройки системы как клиента LDAP с помощью команды **mksecldap** необходимо настроить систему для использования LDAP в качестве домена поиска, на котором будет производиться поиск данных стратегий TSD и TE, задав параметр secorder в файле **/etc/nscontrol.conf**.

После настройки системы в качестве клиента LDAP и домена поиска данных стратегий TSD и TE демон клиента **/usr/sbin/secldapclntd** извлекает данные стратегий TSD и TE из сервера LDAP при каждом выполнении команд **trustchk** на клиенте LDAP.

Включение LDAP с помощью команды trustchk:

Все команды управления базой данных стратегий TSD и TE позволяют использовать базы данных стратегий TSD и TE LDAP.

Выполните команду **trustchk** с флагом **–R** для начальной установки базы данных LDAP. Начальная установка включает в себя добавление стратегий TSD, стратегий TE, базовых DN и создание файлов **/etc/security/tsd/ldap/tsd.dat** и **/etc/security/tsd/ldap/tepolicies.dat** локальной базы данных.

Если команда **trustchk** выполняется с флагом **–R** с помощью опции LDAP, то операции основаны на данных сервера LDAP. Если команда **trustchk** выполняется с флагом **–R** с помощью опции filles, то операции основаны на данных локальной базы данных. По умолчанию для флага **–R** используется опция files.

Информация, связанная с данной:

Команда mksecldap

Команда trustchk

Контроль целостности базы данных надежных сигнатур:

Команду **trustchk** можно использовать для проверки целостности определений файлов в базе данных надежных сигнатур (TSD) по сравнению с реальными файлами.

Если команда **trustchk** выявит несоответствие, то она может либо автоматически исправить его, либо запросить у пользователя разрешение на попытку исправления. В случае несоответствия в размере, сигнатуре, cert_tag или hash_value исправление невозможно. В таких ситуациях команда **trustchk** сделает файл недоступным, переведя ее тем самым в разряд бесполезных и содержащих какие-либо повреждения.

Для различных несоответствующих атрибутов могут быть предприняты следующие меры по исправлению:

owner В качестве владельца файла будет указано значение из TSD.

группа В качестве группы файла будет указано значение из TSD.

mode В качестве флагов режима файла будет указано значение из TSD.

hardlinks
Если ссылка указывает на какой-либо другой файл, то она будет изменена таким образом, чтобы она указывала на этот файл. Если ссылка не существует, то создается новая ссылка, указывающая на этот файл.

symlinks
Так же, как с жесткими ссылками.

type Файл будет сделан недоступным.

size Файл будет сделан недоступным, если это не файл **VOLATILE**.

cert_tag
Файл будет сделан недоступным.

signature
Файл будет сделан недоступным, если это не файл **VOLATILE**.

hash_value
Файл будет сделан недоступным, если это не файл **VOLATILE**.

minslabel
В системе Trusted AIX метке минимальной чувствительности будет присвоено значение из TSD.

maxslabel
В системе Trusted AIX метке максимальной чувствительности будет присвоено значение из TSD.

intlabel
В системе Trusted AIX метке целостности будет присвоено значение из TSD.

accessauths
В качестве прав доступа будет указано значение из TSD. В системе Trusted AIX значения **t_accessauths** считаются частью атрибута **accessauths**.

innateprivs
В качестве изначальных прав доступа будет указано значение из TSD. В системе Trusted AIX значения **t_innateprivs** считаются частью атрибута **innateprivs**.

inheritprivs
В качестве наследуемых прав доступа будет указано значение из TSD. В системе Trusted AIX значения **t_inheritprivs** считаются частью атрибута наследования.

authprivs
В качестве санкционированных прав доступа будет указано значение из TSD. В системе Trusted AIX значения **t_authprivs** считаются частью атрибута **authprivs**.

aecflags
В качестве флагов защиты будет указано значение из TSD. В системе Trusted AIX значения **t_secglags** считаются частью атрибута **secflags**.

Также можно проверить определения файлов с использованием другой базы данных с помощью опции **-F**. Системный администратор должен избегать хранения TSD в той же самой системе. Резервную копию базы данных следует создавать в другом расположении. Таким образом, проверить целостность файлов можно с использованием резервной копии TSD с использованием опции **-F**.

Конфигурация стратегий защиты:

Функция защищенного выполнения (TE) предоставляет механизм проверки целостности файлов во время выполнения. Используя этот механизм, систему можно настроить на проверку целостности защищенных файлов перед каждым запросом на доступ к этому файлу, разрешая его только для тех защищенных файлов, которые проходят проверку целостности.

Когда файл отмечен как защищенный (путем добавления его определения в Базу данных надежных сигнатур), функцию TE можно настроить на проверку его целостности при каждой попытке доступа. TE может постоянно проверять систему и способна выявить подделки любых защищенных файлов (злонамеренным пользователем или вредоносным приложением), которые существуют в системе во время выполнения (например, во время загрузки). При нахождении подделанного файла TE может принять меры по его исправлению на основании заданной стратегии, например, запретить выполнение файла, доступ к файлу или сделать запись в протоколе ошибок. Если файл открыт или выполняется и запись о нем существует в Базе данных надежных сигнатур (TSD), то TE поступает следующим образом:

- Перед загрузкой двоичного файла компонент, ответственный за загрузку файла (системный загрузчик), вызывает подсистему защищенного выполнения и вычисляет значение хэш-функции с помощью алгоритма SHA-256 (настраиваемого).
- Это значение, полученное во время выполнения, сравнивается со значением из TSD.
- Если значения совпадают, то открытие или выполнение файла разрешаются.
- Если значения не совпадают, то двоичный файл либо подделан, либо скомпрометирован. Пользователь самостоятельно решает, какие меры следует принять. Механизм TE предоставляет пользователям опции настройки собственных стратегий относительно мер, принимаемых в случае несовпадения значений хэш-функции.
- На основании этих стратегий конфигурации принимаются соответствующие меры.

Можно применить следующие стратегии:

CHKEXEC
Проверять значение хэш-функции только для защищенных исполняемых файлов перед их загрузкой в память для выполнения.

CHKSHLIBS
Проверять значение хэш-функции только для защищенных общих библиотек перед их загрузкой в память для выполнения.

CHKSCRIPTS
Проверять значение хэш-функции только для защищенных основных сценариев перед их загрузкой в память.

CHKKERNEXT
Проверять значение хэш-функции только для расширений ядра перед их загрузкой в память.

STOP_UNTRUSTD
Остановить загрузку незащищенных файлов. Загружаются только файлы, принадлежащие TSD. Эта

стратегия срабатывает только в совокупности со стратегией CHK*, указанной выше. Например, **если CHKEXEC=ON** и **STOP_UNTRUSTD=ON**, то выполнение запрещается для всех исполняемых двоичных файлов, не принадлежащих TSD.

STOP_ON_CHKFAIL
Остановить загрузку защищенных файлов при несовпадении значений хэш-функции. Эта стратегия также работает в комбинации со стратегией CHK*. Например, если **CHKSHLIBS=ON и STOP_ON_CHKFAIL=ON**, то блокируется загрузка в память всех общих библиотек, не принадлежащих TSD.

TSD_LOCK
Заблокировать TSD, чтобы ее редактирование стало невозможным.

TSD_FILES_LOCK
Заблокировать защищенные файлы. Это не позволит открытие защищенных файлов в режиме записи.

TE Включить/отключить функции защищенного выполнения. Указанные стратегии работают только если эти функции включены.

В следующей таблице описано взаимодействие между различными стратегиями CHK* и STOP*, когда они включены:

Стратегия	STOP_UNTRUSTD	STOP_ON_CHKFAIL
CHKEXEC	Остановить загрузку исполняемых файлов, не принадлежащих TSD.	Остановить загрузку исполняемых файлов, если их значения хэш-функций не совпадают со значениями в TSD.
CHKSHLIBS	Остановить загрузку общих библиотек, не принадлежащих TSD.	Остановить загрузку общих библиотек, если их значения хэш-функций не совпадают со значениями в TSD.
CHKSCRIPTS	Остановить загрузку основных сценариев, не принадлежащих TSD.	Остановить загрузку основных сценариев, если их значения хэш-функций не совпадают со значениями в TSD.
CHKKERNEXT	Остановить загрузку расширений ядра, не принадлежащих TSD.	Остановить загрузку расширений ядра, если их значения хэш-функций не совпадают со значениями в TSD.

Примечание: Стратегию можно включать и выключать в любой момент, пока включено TE, благодаря которому действуют стратегии. Когда стратегия действует, ее выключение вступает в силу только в следующем цикле загрузки. Все информационные сообщения записываются в **syslog**.

Информация, связанная с данной:

TE_verify_reg kernel service

TE_verify_unreg Kernel Service

Путь защищенного выполнения и защищенной библиотеки:

Путь защищенного выполнения (TEP) определяет список каталогов, которые сдержат защищенные исполняемые файлы. При включенной проверке TEP системный загрузчик разрешает выполнение только тех двоичных файлов, которые расположены по указанным путям. Путь защищенной библиотеки (TLP) выполняет те же функции, но используется для указания каталогов, содержащих защищенные библиотеки системы.

При включенном TLP системный загрузчик разрешает подключение к двоичным файлам только для тех библиотек, которые расположены по указанным путям. Для включения и выключения TEP или TLP, а также их списков путей, разделенных двоеточиями, можно использовать команду **trustchk** с указанием аргументов командной строки TEP и TLP.

Защищенная оболочка и защищенная клавиша внимания:

Защищенная оболочка и защищенная клавиша внимания (SAK) действуют подобно защищенной базе вычислений (TCB) за исключением случаев, когда вместо TCB в системе активировано защищенное выполнение. В этих случаях защищенная оболочка запускает на выполнение только те файлы, которые указаны в базе данных надежных сигнатур.

Для просмотра подробных сведений о TCB и SAK смотрите Защищенная компьютерная база, Использование защищенной клавиши внимания и Настройка защищенной клавиши внимания.

База данных стратегий Надежного выполнения (TE):

Стратегии Надежного выполнения (Trusted Execution (TE)) хранятся в файле **/etc/security/tsd/tepolicies.dat**. Путь для стратегий TE указан с помощью каталогов TLP и каталогов TEP.

Профайл защиты с уровнем оценки 4+, Защита AIX с использованием меток и уровень оценки 4+

Системные администраторы могут установить систему с опциями Базовая защита AIX (BAS) и Уровень оценки 4+ (EAL4+) или Защита AIX на основе меток (LAS) и уровнем оценки 4+ (EAL4+) во время установки базовой операционной системы (BOS). При этом накладывается ряд ограничений на программное обеспечение, устанавливаемое при установке BOS, а также ряд ограничений на сетевой доступ.

Примечание: Оценка AIX версии 7.1 продолжается. Последние данные приведены в Информации о выпуске AIX версии 7.1.

Обзор профайла защиты:

Профайл защиты - это продукт, который устанавливает требования к защите операционных систем общего назначения в сетевых средах. Этот профайл устанавливает требования, необходимые для достижения целей функции защиты "Цель оценки" (TOE) и ее среды.

Профайл защиты содержит базовый пакет и несколько расширенных пакетов. Продукты, которые связаны с поддержкой базового пакета профайла защиты, - это идентификация и проверка прав доступа, избирательный контроль доступа (DAC), контроль, криптографические службы, управление механизмами защиты и связь по защищенному каналу. В профайл защиты включены дополнительные необязательные пакеты для защиты с использованием меток, проверки целостности, расширенного контроля, общей криптографии, расширенного управления, расширенной идентификации и проверки прав доступа, защищенной загрузки и виртуализации.

Допущения

- Среда для использования с TOE:

  Все допущения в разделе относятся к Базовая защита AIX (режим BAS) и Защита AIX на основе меток (режим LAS), если не указано иное. Все допущения относительно виртуального сервера ввода-вывода (VIOS) явно помечены как предназначенные только для VIOS. VIOS не разделяет допущений с операционной системой AIX и с защищенной AIX.
- Физические:

  ИТ-среда предоставляет TOE с соответствующей физической защитой, сопоставимой с ценностью ИТ-ресурсов, защищенных TOE.

  Примечание: Только VIOS: операционная среда предоставляет TOE с соответствующей физической защитой, сопоставимой с ценностью ИТ-ресурсов, защищенных TOE.
- Администрирование:
  – Функциями защиты TOE управляет один или несколько специалистов. Персонал, защищающий систему, не является невнимательным, преднамеренно небрежным или враждебным и придерживается инструкций из руководства.

- Идентифицированные пользователи могут получать доступ к некоторому объему информации, которой управляет TOE, и от них ожидается сотрудническое отношение.
- Пользователи достаточно обучены и надежны для выполнения некоторых задач или групп задач в защищенной ИТ-среде. Они должны иметь полный контроль над своими пользовательскими данными.
- Только VIOS: функциями защиты TOE управляет один или несколько специалистов. Персонал, защищающий систему, не является невнимательным, преднамеренно небрежным или враждебным и придерживается инструкций из руководства.
- Только VIOS: идентифицированные пользователи имеют права получить доступ по крайней мере к некоторому объему информации, которой управляет TOE, и от них ожидается сотрудническое отношение.
- Только VIOS: пользователи достаточно обучены и надежны для выполнения некоторых задач или групп задач в защищенной рабочей среде. Они должны иметь полный контроль над своими пользовательскими данными.

- Процедурные:
  - Любое изменение или искажение файлов TOE, требуемых для защиты или относящихся к защите, со стороны пользователя или базовой платформы, совершенное преднамеренно или непреднамеренно, должно обнаруживаться администратором.
  - Все удаленные ИТ-системы, защищенные с помощью Target Security Function (TSF) с целью предоставления данных или служб TSF для TOE или с целью поддержки TSF в применении решений стратегии защиты, считаются находящимися под одним и тем же управленческим контролем и работающими в ограничениях стратегии защиты, совместимыми со стратегией защиты TOE.
  - Считается, что все удаленные защищенные ИТ-системы, которые защищены с помощью TSF с целью предоставления данных или служб TSF для TOE или с целью поддержки TSF в применении решений стратегии защиты, правильно реализовывают функции, используемые TSF в соответствии с допущениями для этой функции.
  - Обеспечивается целостность следующей информации:
    - Весь код TSF, включая функцию проверки целостности, которая загружается и выполняется перед запуском механизма проверки целостности
    - Все данные TSF, включая данные TSF для проверки целостности, которые используются кодом TSF, загружаемым и запускаемым перед запуском механизма проверки целостности
  - Только VIOS: любое изменение или искажение файлов TOE, требуемых для защиты или относящихся к защите, со стороны пользователя или базовой платформы, совершенное преднамеренно или непреднамеренно, должно обнаруживаться администратором.
- Установка соединений: все входящие и исходящие соединения с удаленными защищенными ИТ-системами и между физически разрозненными частями TSF, которые не защищены с помощью собственно TSF, физически или логически защищены в среде TOE для обеспечения целостности и конфиденциальности передаваемых данных и для гарантии подлинности конечных точек.

Получение программного обеспечения

Для получения программного обеспечения выполните следующие действия:

1. Загрузите продукт.
2. Откройте пункт Справка в меню Поддержка разрешенного программного обеспечения в левой панели. Конфигурация общих критериев требует получения продукта и любых обновлений на физическом носителе или с помощью администратора загрузок.

Информация об установке продукта приведена в разделе Установка системы BAS /EAL4+.

Установка системы BAS/EAL4+:

RBAC автоматически включается при выборе этой опции.

Для выбора опции BAS/EAL4+ при установке BOS выполните следующие действия:

1. В меню Установка и настройка выберите **Дополнительные опции**.
2. В разделе Дополнительные опции выберите **Да** для опции BAS/EAL4+, и если используется WPAR, выберите **Нет** для опции TCB. Если используется настраиваемый файл `bosinst.data` для установки без вывода сообщений, для опции TCB можно выбрать **Да**.

Для установки BAS запретите удаленный вход в систему под именем root. Для того чтобы запретить удаленный вход в систему под именем root, после установки выполните следующую команду:

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

Добавьте администраторов в группу **SUADMIN**, чтобы они могли выполнять команду **su** для root.

Опция **Включить поддержку технологии BAS и EAL4+** доступна лишь при выполнении следующих условий:

- Выбрана установка с заменой всех данных.
- Выбран английский язык.
- Включена поддержка 64-разрядного ядра.
- Включена поддержка расширенной журнализированной файловой системы (JFS2).

Если для опции **Включить поддержку технологии BAS и EAL4+** задано значение Да, то для опции **Защищенная компьютерная база** также должно быть выбрано значение Да, а для опции **Рабочий стол** можно указать только значение Нет или CDE.

Для выполнения установки без вывода сообщений с использованием настраиваемого файла `bosinst.data` в поле **INSTALL_TYPE** должно быть указано значение `CC_EVAL`, а следующие поля должны быть заданы следующим образом:

```
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
```

Дополнительные сведения о RBAC приведены в разделе Контроль Доступа, основанный на ролях (RBAC).

Среда управления сетевой установкой для BAS/EAL4+:

Клиентов BAS/EAL4+ можно устанавливать с помощью среды управления сетевой установкой (NIM).

Мастер NIM настраивается на предоставление ресурсов, необходимых для установки соответствующего BAS/EAL4+ уровня AIX 7.1. После этого можно установить клиентов NIM с помощью ресурсов, расположенных на сервере NIM. Можно выполнить установку NIM клиента без вывода сообщений, задав следующие поля в ресурсе **`bosinst_data`**:

```
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
```

Сервер NIM нельзя настроить в качестве системы BAS/EAL4+ и нельзя использовать в одной сети с другими системами BAS/EAL4+. При инициализации установки с сервера NIM, пункт меню **Оставить клиент NIM после установки SMIT** должен установлен в значение "Нет". После установки клиента NIM в качестве системы BAS/EAL4+ необходимо удалить этот клиент из сети сервера NIM. Это приведет к невозможности установки и обновления программного обеспечения клиента с помощью данного сервера NIM.

В качестве примера можно рассмотреть две сети: первая включает сервер NIM и системы, не отвечающие требованиям BAS/EAL4+; а вторая - только системы BAS/EAL4+. Выполните установку клиента NIM с помощью NIM. После завершения установки отключите только что установленные системыBAS/EAL4+ от сети сервера NIM и подключите их к тестовой сети.

Второй пример включает только одну сеть. Сервер NIM не подключен к сети в то время как остальные системы работают в тестовой конфигурации, а системы BAS/EAL4+ не подключены к сети во время установки NIM.

Комплект программного обеспечения BAS/EAL4+:

При выборе опции **BAS/EAL4+** устанавливается содержимое комплекта программного обеспечения `/usr/sys/inst.data/sys_bundles/CC_EVAL.BOS.autoi`.

При выборе опции **BAS/EAL4+** можно также дополнительно установить комплект графического программного обеспечения и комплект службы поиска документации. Если вы выбрали опцию **BAS/EAL4+** и решили установить **Графическое программное обеспечение**, то будет установлен комплект `/usr/sys/inst.data/sys_bundles/CC_EVAL.Graphics.bnd`. Если вы выбрали опцию **BAS/EAL4+** и решили установить программное обеспечение службы поиска документации, то будет установлен комплект `/usr/sys/inst.data/sys_bundles/CC_EVAL.DocServices.bnd`.

После установки лицензионных программ (LPP) конфигурация системы по умолчанию изменяется согласно требованиям BAS/EAL4+. В стандартную конфигурацию вносятся следующие изменения:

- Из файла `/etc/pse.conf` удаляется запись `/dev/echo`.
- Создаются экземпляры потоковых устройств.
- Доступ к съемным носителям разрешается только пользователю root.
- Из файла `inetd.conf` удаляются все записи, не относящиеся к CC.
- Изменяются права доступа к различным файлам.
- В файле `sysck.cfg` регистрируются символьные связи.
- В файле `sysck.cfg` регистрируются устройства.
- Устанавливаются атрибуты по умолчанию для пользователей и портов.
- Приложение doc_search настраивается для работы с браузером.
- Из файла `inittab` удаляется запись httpdlite.
- Из файла `inittab` удаляется запись writesrv.

- Из файла inittab удаляется запись mkatmpvc.
- Из файла inittab удаляется запись atmsvcd.
- В файле /etc/rc.tcpip отключается запись snmpd.
- В файле /etc/rc.tcpip отключается запись hostmibd.
- В файле /etc/rc.tcpip отключается запись snmpmibd.
- В файле /etc/rc.tcpip отключается запись aixmibd.
- В файле /etc/rc.tcpip отключается запись muxatmd.
- Порт NFS (2049) объявляется привилегированным.
- В файл /etc/security/audit/events добавляются дополнительные события.
- Проверяется работоспособность интерфейса loopback.
- Создаются синонимы /dev/console.
- Принудительно устанавливаются права доступа по умолчанию для соединений с X-сервером.
- Для каталога /var/docsearch устанавливаются права доступа на чтение для всех пользователей.
- В ODM добавляются разделы, задающие права доступа к консоли.
- Для терминалов BSD устанавливается режим доступа 000.
- Отключаются файлы .netrc.
- Добавляются процедуры обработки каталога исправлений.

Графический пользовательский интерфейс:

Совместимая система BAS/EAL4+ включает X Windows System в качестве графического пользовательского интерфейса.

X Window реализует механизм показа графических приложений (часы, калькуляторы и т. п.) и сеансов терминала (команда **aixterm**). X Window запускается командой **xinit**.

Запуск сеанса X Window:

```
xinit
```

Эта команда запускает сервер X Window, к которому разрешен только локальный доступ и только пользователю, запустившему сервер. Клиенты X Window с правами root смогут получить доступ к этому серверу X Window через сокет UNIX. Все остальные клиенты доступа к данному серверу X Window иметь не будут. Это ограничение предотвращает получение несанкционированного доступа к серверу.

Установка системы LAS/EAL4+:

RBAC автоматически включается при выборе этой опции.

Для выбора опции LAS/EAL4+ при установке BOS выполните следующие действия:

Опции установки можно выбрать, введя 3, чтобы сменить **Модель защиты** и 4 для просмотра поля **Дополнительные опции** в окне Способ установки и параметры системы. Эти опции различны для различных типов установки (замена всех данных, сохранение или обновление) и опций защиты. Для LAS методом установки является новая установка или полное переопределение. Выберите опцию **Установка конфигурации LAS/EAL4+**.

Дополнительные сведения о RBAC приведены в разделе Контроль Доступа, основанный на ролях (RBAC).

Установка конфигурации LAS/EAL4+ (только с Trusted AIX):

Опция установки конфигурации **LAS/EAL4+** устанавливает Trusted AIX в режиме конфигурации LAS/EAL4+. Режим конфигурации LAS/EAL4+ обеспечивает еще более строгую защиту по сравнению с режимом установки Trusted AIX.

Если вы выполняете автономную установку с помощью пользовательского файла `bosinst.data`, то в поле **INSTALL_TYPE** должно быть указано пустое значение, а в поле**TRUSTED_AIX** - значение `yes`, а следующие поля должны иметь указанные ниже значения:

```
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
```

Дополнительная информация о Trusted AIX приведена в разделе Trusted AIX.

Среда управления сетевой установкой для LAS/EAL4+:

Клиентов LAS/EAL4+ можно устанавливать с помощью среды управления сетевой установкой (NIM).

Мастер NIM настраивается на предоставление ресурсов, необходимых для установки соответствующего LAS/EAL4+ уровня AIX 7.1. После этого можно установить клиентов NIM с помощью ресурсов, расположенных на сервере NIM. Можно выполнить установку NIM клиента без вывода сообщений, задав следующие поля в ресурсе bosinst_data:

```
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
```

Сервер NIM нельзя настроить в качестве системы LAS/EAL4+ и нельзя использовать в одной сети с другими системами LAS/EAL4+. При инициализации установки с сервера NIM, пункт меню **Оставить клиент NIM после установки SMIT** должен установлен в значение "Нет". После установки клиента NIM в качестве системы LAS/EAL4+ необходимо удалить этот клиент из сети сервера NIM. Это приведет к невозможности установки и обновления программного обеспечения клиента с помощью данного сервера NIM.

В качестве примера можно рассмотреть две сети: первая включает сервер NIM и системы, не отвечающие требованиям LAS/EAL4+; а вторая - только системы LAS/EAL4+. Выполните установку клиента NIM с помощью NIM. После завершения установки отключите только что установленные системыLAS/EAL4+ от сети сервера NIM и подключите их к тестовой сети.

Второй пример включает только одну сеть. Сервер NIM не подключен к сети в то время как остальные системы работают в тестовой конфигурации, а системы LAS/EAL4+ не подключены к сети во время установки NIM.

Физическая среда систем BAS/EAL4+ and LAS/EAL4+:

Системы BAS/EAL4+ и LAS/EAL4+ предъявляют особые требования к среде, в которой будут работать.

Эти требования перечислены ниже:

- Физический доступ к системе должен быть ограничен, чтобы с системными консолями могли работать только администраторы, имеющие соответствующий уровень доступа.
- Служебный процессор не должен быть подключен к модему.
- Физический доступ к терминалам должен быть разрешен только пользователями с соответствующим уровнем доступа.
- Физические компоненты сети должны обеспечивать защиту от перехвата информации и от несанкционированного доступа путем имитации (т.е. от троянских коней). При использовании незащищенных линий связи должны применяться дополнительные меры защиты, например, шифрование.
- Связь с другими системами, не являющимися системами AIX 7.1 BAS/EAL4+ или LAS/EAL4+, либо находящимися в другой сфере управления, не допускается.
- Для связи с другими системами BAS/EAL4+ и LAS/EAL4+ необходимо использовать только IPv4. IPv6 включается в конфигурацию, при этом в нее входят лишь те возможности IPv6, которые поддерживаются IPv4.
- Пользователям должно быть запрещено изменять системное время.
- Системы в среде LPAR не могут использовать одни и те же PHB.

Организационная среда систем BAS/EAL4+ и LAS/EAL4+:

Для систем BAS/EAL4+ и LAS/EAL4+ должны быть удовлетворены определенные процедурные и организационные требования.

Требования следующие:

- Администратор должен пройти обучение.
- ИД пользователей должны создаваться только для пользователей, допущенных к работе с информацией, хранящийся в этих системах.
- Пользователи должны применять надежные пароли (случайные последовательности символов, не связанные с самим пользователем или с организацией). Информация о настройке правил проверки паролей приведена в разделе “Пароль” на стр. 64.
- Пользователи ни при каких условиях не должны передавать свои пароли другим лицам.
- Уровень знаний администраторов должен быть достаточно высоким и обеспечивающим эффективное управление системами, критичными к уровню защиты.
- Администраторы должны работать в соответствии с рекомендациями, приведенными в документации по системе.

- Администраторы должны входить в системы под своим личным ИД пользователя, а затем командой **su** переключаться в режим администратора для выполнения соответствующих задач.
- Пароли, сформированные администраторами для пользователей системы, должны передаваться пользователям с соблюдением строгих мер безопасности.
- Сотрудники, ответственные за организацию работы системы, должны разработать и реализовать процедуры, обеспечивающие безопасную работу.
- Администраторы должны обеспечить ограничение доступа к критически важным системным ресурсам с помощью соответствующих битов режима доступа или ACL.
- Физическая сеть должна быть сертифицирована для передачи наиболее конфиденциальных данных, хранящихся в системе.
- Процедуры обслуживания должны включать регулярную диагностику системы.
- Администраторы должны располагать планом процедур обеспечения надежной работы и восстановления после аварии.
- Не следует изменять переменную среды *LIBPATH*, поскольку в противном случае будет возможна загрузка незащищенной библиотеки защищенным процессом.
- В операционной системе не должно применяться программное обеспечение трассировки и низкоуровневого мониторинга передаваемых данных (tcpdump, trace и т.д.).
- Протоколы с анонимным доступом, например, HTTP, могут применяться только для передачи общедоступной информации (например, электронной документации).
- Следует применять только NFS на основе TCP.
- У пользователей не должно быть доступа к съемным носителям. Файлы устройств должны быть защищены с помощью соответствующих битов режима доступа или ACL.
- Администраторы не должны применять функции динамического управления разделами для выделения и отключения ресурсов. Настройку разделов можно выполнять только в том случае, если работа всех разделов завершена.

BAS/EAL4+ и среда операционной системы LAS/EAL4+:

Для BAS/EAL4+ и системы LAS/EAL4+ должны быть удовлетворены определенные операционные требования и процедуры.

Требования следующие:
- При использовании Консоль аппаратного обеспечения (HMC), система HMC должна находиться в физически управляемой среде.
- Доступ к рабочей среде и HMC должен быть только у уполномоченных сотрудников.
- HMC можно применять только для выполнения следующих задач:
  - Первоначальная настройка разделов. Во время настройки разделы активировать нельзя.
  - Перезапуск "зависших" разделов.
- HMC нельзя применять в ходе работы настроенных систем.
- Системная функция вызова сервисного центра должна быть отключена.
- Удаленный доступ к системе с помощью модема должен быть запрещен.
- Если AIX работает в среде с поддержкой LPAR, то администратор должен обеспечить выполнение требований EAL4+ при использовании LPAR для работы с логическими разделами.
- В логических разделах функция служебных прав доступа должна быть отключена.

Настройка системы BAS/EAL4+:

Можно настроить систему Базовая защита AIX (BAS) и Уровень оценки 4+ (EAL4+).

Группы **system**, **sys**, **adm**, **uucp**, **mail**, **security**, **cron**, **printq**, **audit** и **shutdown** являются административными группами. В эти группы можно добавлять только надежных пользователей.

Администрирование:

Администраторы должны входить в систему под управлением собственной учетной записи, а затем с помощью команды **su** переключаться на учетную запись root для выполнения функций управления системой.

Для того чтобы обеспечить защиту от угадывания пароля пользователя root, доступ к команде **su** для переключения на учетную запись root должен быть разрешен только администраторам. Для этого выполните следующие действия:

1. Добавьте запись в раздел **root** файла `/etc/security/user`:
   ```
   root:
     admin = true
     .
     .
     .
     sugroups = SUADMIN
   ```
2. В файле `/etc/group` должна быть определена группа, содержащая только ИД пользователей администраторов, например:
   ```
   system:!:0:root,paul
   staff:!:1:invscout,julie
   bin:!:2:root,bin
   .
   .
   .
   SUADMIN:!:13:paul
   ```

Администраторы должны также строго следовать следующим правилам:

- Разработать и реализовать процедуры, гарантирующие безопасное распространение, установку и настройку входящих в состав системы аппаратных и программных компонентов.
- Настроить систему таким образом, чтобы устанавливать защищенное программное обеспечение могли только администраторы.
- Реализовать процедуры очистки экрана перед выходом пользователей терминала из системы (например, терминала IBM® 3151).

Конфигурация пользователей и портов:

Опции настройки пользователей и портов AIX должны отвечать требованиям, предъявляемым при оценке. Фактическим требованием является то чтобы TSF предоставляла механизм правильного предположения пароля, который удовлетворяет показателю качества. Вероятность правильного предположения пароля, который может быть получен от атакующего во время срока действия пароля, должна быть меньше, чем 2^-20.

В показанном ниже примере файла `/etc/security/user` используется словарный список `/usr/share/dict/words`. Файл `/usr/share/dict/words` входит в состав набора файлов `bos.data`. Перед настройкой файла `/etc/security/user` необходимо установить набор файлов `bos.data`. Ниже перечислены рекомендуемые значения параметров в файле `/etc/security/user`:

```
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
```

Значения по умолчанию, указанные в файле `/etc/security/user`, не должны переопределяться значениями, заданными для отдельных пользователей.

Примечание: Указание значения `login = false` а разделе root запрещает пользователю root непосредственный вход в систему. Работать под управлением учетной записи root смогут только пользователи, имеющие права на переключение на учетную запись root с помощью **su**. В случае атаки системы путем отправки множества запросов на вход в систему с указанием неправильных паролей, возможно блокирование всех учетных записей пользователей. Такая атака может привести к невозможности входа в систему любых пользователей, включая администраторов. После блокирования учетной записи пользователь не сможет войти в систему до тех пор, пока администратор не сбросит атрибут `unsuccessful_login_count` в файле `/etc/security/lastlog`, присвоив ему значение, меньшее, чем значение атрибута `loginretries` для этого пользователя. В случае блокировки всех учетных записей администраторов, может потребоваться перезагрузить систему в режиме обслуживания и запустить команду **chsec**. Дополнительная информация о команде **chsec** приведена в разделе “Управление учетными записями пользователей” на стр. 53.

Ниже перечислены рекомендуемые значения параметров в файле `/etc/security/login.cfg`:

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

Список программ setuid/setgid:

Для систем AIX с BAS создается список защищенных приложений.

Разряды **suid/sgid** отключены для всех незащищенных программ, которые принадлежат учетной записи root, или защищенной группе. После установки BAS в системе есть только следующие программы: **system**, **sys**, **adm**, **uucp**, **mail**, **security**, **cron**, **printq**, **audit** и **shutdown**. Это программы **suid**, принадлежащие учетной записи root и программы **sgid**, принадлежащие этим защищенным группам. Просто добавьте защищенных пользователей в эти группы.

В список защищенных приложений попадают все приложения, относящиеся хотя бы к одной из следующих категорий:

- Для соответствующего приложения включен разряд root SUID
- Для одной из защищенных групп включен разряд SGID
- Приложения, имеющие доступ к каким-либо защищенным базам данных, в соответствии с документацией администратора

Примечание: Бит **setuid** команды **ipcs** должен быть удален системным администратором. Системный администратор должен запустить команды **chmod u-s /usr/bin/ipcs** и **chmod u-s /usr/bin/ipcs64**.

Изменение файловой системы контроля:

RBAC автоматически включается при выборе этой опции.

Файловой системой `/audit` является файловая система `jfs`. Ее необходимо заменить на файловую систему `jfs2`. Кроме того, системы BAS требуют дополнительных команд. Для внесения изменений в файловую систему выполните следующее:

1. Для изменения файловой системы для BAS введите команду:

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

   Для систем LAS выполните этап 3.
2. Если поле TYPE содержит символ вопросительного значка (?), введите команду:

   ```
   synclvodm -v rootvg
   ```
3. Удалите файловую систему `jfs` и создайте файловую систему `jfs2` с помощью команды:

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

Обновление базы данных сигнатуры защиты (TSD):

В этом разделе описывается процедура обновления TSD.

Настройка BAS/LAS изменяет системные режимные биты, наблюдаются ошибки целостности TSD.

Во время перезагрузки системы выберите опцию **Игнорировать все**.

Для обновления TSD введите команду:

```
trustchk -u ALL mode
```

Использование системы LAS:

В этом разделе даются рекомендации по использованию системы LAS.

После установки системы как `isso` задайте для опции автоматической перезагрузки значение **false** вводом команды:

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

Если TSD продолжает выдавать ошибки `intlabel`, удалите ошибки с помощью `isso` с правами доступа **PV_ROOT** вводом команды:

```
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
```

Если сообщения об ошибках, связанных с контролем, выводятся на консоль, то используйте права доступа `isso` для перезапуска системы контроля вводом команд:

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

После трех неудачных попыток входа в систему вход в систему `isso/so` блокируется сетью. Однако администратор может продолжить попытки входа с этими учетными записями на локальной консоли.

Результат выполнения команд `cron/at` не направляется в буфер почты пользователя.

Всемирные каталоги, которые имеют диапазоны меток (пример: `/tmp`), не разбиты на разделы. Для предотвращения возможности протекания информации между метками администратор должен разбить на разделы эти каталоги сразу после первоначальной настройки.

Сетевой интерфейс:

В этом разделе описывается процедура использования сетевого интерфейса.

В Trusted AIX сетевой интерфейс по умолчанию имеет некоторый диапазон меток `minSL=impl_lo` и `maxSL=ts_all`. В системах LAS/EAL4+ нет диапазона меток. Правило по умолчанию автоматически изменяется на `impl_lo` при выборе опции установки LAS/EAL4+. Для изменения правила по умолчанию на `isso` используйте команду **netrule**.

Например:

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

Обновление WPAR:

В этом разделе описывается процедура создания рабочих разделов (WPAR) для AIX, совместимых с EAL4+.

Создайте WPAR в BAS системе и выполните следующую команду в WPAR для совместимости с EAL4+:

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

При выполнении `clogin` в системе LAS впервые выполняются сценарии начальной загрузки (включая `CC_EVALify.sh`).

Сценарии начальной загрузки приводят к более длительному выполнению `clogin`, чем обычно, когда `clogin` вызывает TSM для входа в систему. Однако WPAR еще находится в режиме настройки, поэтому вход в систему отклоняется. Необходимо подождать приблизительно 10 минут завершения настройки WPAR перед другой попыткой `clogin`. Для вновь созданных систем WPAR опции пользователя по умолчанию должны быть заданы в соответствии с требованиями оценки, которые включают:

- `root` в режиме BAS
- `isso/sa/so` в режиме LAS

Пользователи `root` и `isso` не имеют пароля или для них требуется лишь слабый пароль. Пароли необходимо обновить перед возможностью доступа другого пользователя к глобальной среде или соответствующему WPAR.

Требованием оценки к паролю является, чтобы вероятность правильного угадывания пароля составляла по крайней мере 1 к 1 000 000, и вероятность правильного угадывания пароля при повторных попытках в течение одной минуты составляла по крайней мере 1 к 100 000. В соответствии с этим требованием параметры пользователя в файле `/etc/security/user` изменяются следующим образом:

```
по умолчанию:
 maxage       = 8
 maxexpired   = 1
 minother     = 2
```

```
minlen       = 8
maxrepeats   = 2
loginretries = 3
histexpire   = 52
histsize     = 20
```

Обновление EFS:

В этом разделе описывается процедура задания атрибутов защиты EFS, которая была оценена как шифровальная файловая система.

Оценка не включает аспекты режима защиты root в сравнении с полным доступом `root`. При включении EFS задайте атрибуты защиты для команд **efsmgr** и **egskeymgr** выполнением команды:

```
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
```

Очистка жесткого диска:

AIX позволяет очищать жесткие диски с помощью сервисного средства **Форматирование носителей** в пакете диагностики AIX. Пакет диагностики подробно описан в книге *Диагностическая информация для систем с несколькими шинами*, и в руководстве пользователя аппаратного обеспечения.

Для того чтобы очистить жесткий диск, введите следующую команду:

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

Эта команда запускает сервисное средство **Форматирование носителей** в интерфейсе, управляемом с помощью меню. При необходимости укажите свой терминал.

Выводится список выбора ресурсов. Выберите жесткие диски, которые вы хотите удалить из этого списка и подтвердите свой выбор, следуя инструкциям на экране.

После подтверждения выбора, выберите в меню **Очистить диск**. Вам будет предложено подтвердить свой выбор. Выберите **Да**.

Вам будет задан вопрос, хотите ли вы **Прочесть информацию с диска** или **Записать шаблоны на диск**. Выберите **Записать шаблоны на диск**.

Далее у вас будет возможность изменить опции очистки диска. После указания необходимых опций, выберите **Подтвердить изменения**. Запущена очистка диска.

Примечание: Выполнение этого процесса может занять большое количество времени.

Ограничения на ресурсы:

Задавая ограничения для ресурсов в файле `/etc/security/limits`, убедитесь, что установленные ограничения соответствуют требованиям процессов.

В частности, никогда не назначайте размер `stack` равным `unlimited`. Стек неограниченного размера может перезаписать другие сегменты выполняющегося процесса. Размер `stack_hard` также нужно ограничить.

Подсистема контроля:

Существует ряд процедур, с помощью которого можно обеспечить безопасность подсистемы контроля.

- Настройте подсистему контроля для записи всех пользовательских действий, относящихся к защите. При этом необходимо выделить для контрольных данных отдельную файловую систему, чтобы обеспечить необходимый объем свободного пространства и избежать конкуренции за дисковое пространство с другими приложениями и данными.
- Защитите контрольные данные (например, контрольные следы, приемные лотки и другие данные в `/audit`) от пользователей, отличных от root.
- В системе BAS/EAL4+ подсистема контроля должна работать в режиме **лотка**. Информация о настройке режимов контроля приведена в разделе “Настройка контроля” на стр. 149.
- Для хранения контрольного журнала должно быть выделено не менее 20 процентов доступного дискового пространства системы.
- Если контроль включен, то параметру *binmode* в разделе start файла `/etc/security/audit/config` должно соответствовать значение `panic`. Параметр *freespace* в разделе bin должен иметь значение, равное не менее 25% от дискового пространства, выделенного для хранения контрольного журнала. Параметрам *bytethreshold* и *binsize* должно соответствовать значение 65 536 байт.
- С целью архивации копируйте контрольные записи из системы на постоянные носители.

Отдельные файлы в распределенной системе:

Следующие файлы распределенной системы, расположенные в каталоге `/etc/security`, не являются общими, а задают параметры отдельных хостов:

/etc/security/failedlogin
: Файл протокола неудачных попыток входа в систему данного хоста.

/etc/security/lastlog
: Информация о успешных и неудачных попытках входа пользователей в систему данного хоста.

/etc/security/login.cfg
: Информация о параметрах входа в систему данного хоста, включая защищенный путь, оболочки входа в систему и т.д.

/etc/security/portlog
: Информация о блокировках портов данного хоста.

Автоматически создаваемые резервные копии общих файлов также не являются общими. Имя файла резервной копии совпадает с именем исходного файла с префиксом `o`.

Применение функции DACinet для управления доступом к сети на уровне пользователей и портов:

Функция DACinet позволяет ограничивать доступ пользователей к функциям TCP.

Дополнительная информация о DACinet приведена в разделе “Управление доступом к порту TCP на уровне пользователей и самостоятельный контроль доступа к портам Internet” на стр. 215. Например, если с помощью DACinet ограничить доступ к порту TCP/25 для входящих соединений только с поддержкой DACinet, то к этому порту смогут обращаться только пользователи root с хостов BAS/EAL4+. Такая конфигурация обеспечивает защиту от фальсификации сообщений электронной почты другими пользователями путем подключения с помощью telnet к порту TCP/25 компьютера жертвы.

Для активации ACL для соединений TCP при загрузке в `/etc/inittab` запускается сценарий `/etc/rc.dacinet`. Этот сценарий считывает определения из файла `/etc/security/acl` и загружает ACL в ядро. Порты, которые не должны защищаться с помощью ACL, должны быть перечислены в `/etc/security/services`.

Если все подключенные системы находятся в подсети 10.1.1.0/24, то для разрешения доступа к X (TCP/6000) только пользователю root необходимо указать в файле `/etc/security/acl` следующую запись ACL:

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

Установка дополнительного программного обеспечения в системе BAS/EAL4+:

Администратор может установить в системе BAS/EAL4+ дополнительное программное обеспечение. Если это ПО работает не под управлением учетной записи пользователя root и не использует права доступа пользователя root, то его установка не будет противоречить спецификации BAS/EAL4+. В качестве примера можно привести офисные приложения, с которыми могут работать обычные пользователи, и которые не используют компоненты SUID.

Установка программного обеспечения, использующего полномочия root, является нарушением требований спецификации BAS/EAL4+. Это значит, например, что нельзя устанавливать старые драйверы JFS, поскольку они работают в режиме ядра. Все приложения, предоставляемые с одним или более правами доступа через `/etc/security/privcmds`, недопустимы. Дополнительные демоны, работающие под управлением учетной записи root (например, демон SNMP), также являются нарушением спецификации BAS/EAL4+. Активизированная система BAS/EAL4+ обычно не подлежит обновлению.

Системы BAS/EAL4+ редко используются именно в той конфигурации, в которой проводились испытания. Особенно это верно для коммерческих сред. Как правило требуется установка дополнительных служб, поэтому рабочая система основывается на сертифицированной, но не соответствует в точности спецификации сертифицированной системы.

Стратегия управления содержимым и списками управления доступа NFS v4:

Список управления доступом NFS v4 (ACL) содержит поля **Тип**, **Маска**, и **Флаги**.

Ниже приведено описание этих полей:

- Поле **Type** содержит одно из следующих значений:
  - `ALLOW` – Предоставляет субъекту, указанному в поле **Кто**, права доступа, указанные в поле **Маска**.
  - `DENY` – Запрещает для субъекта, указанного в поле **Кто**, права доступа, указанные в поле **Маска**.
- Поле Маска содержит одно или более значений, четко определяющих права доступа:
  - `READ_DATA / LIST_DIRECTORY` – Чтение данных из объекта, который не является каталогом, или просмотр объектов в каталоге.
  - `WRITE_DATA / ADD_FILE` – Запись данных в объект, который не является каталогом, или добавление в каталог объекта, который не является каталогом.
  - `APPEND_DATA / ADD_SUBDIRECTORY` – Добавление данных в объект, который не является каталогом, или добавление в каталог вложенного каталога.
  - `READ_NAMED_ATTRS` – Чтение именованных атрибутов объекта.
  - `WRITE_NAMED_ATTRS` – Запись именованных атрибутов объекта.
  - `EXECUTE` – Выполнение файла или проход/поиск по каталогу.
  - `DELETE_CHILD` – Удаление из каталога файла или каталога.
  - `READ_ATTRIBUTES` – Чтение базовых (не относящихся к ACL) атрибутов файла.
  - `WRITE_ATTRIBUTES` – Изменение атрибутов времени, связанных с файлом или каталогом.
  - `DELETE` – Удаление файла или каталога.
  - `READ_ACL` – Чтение ACL.
  - `WRITE_ACL` – Запись ACL.
  - `WRITE_OWNER` – Изменение пользователя и группы.
  - `SYNCHRONIZE` – Синхронизация доступа (существует для совместимости с другими клиентами NFS v4, но ее функциональность не реализована).
- Поле **Флаги** – Это поле определяет возможности ACL каталога относительно наследования и указывает, содержит ли поле **Кто** группу. Это поле может быть пустым или содержать один и более следующих флагов:

  - **FILE_INHERIT** – Указывает, что в этом каталоге новые объекты, которые не являются каталогами, наследуют эту запись.
  - **DIRECTORY_INHERIT** – Указывает, что в этом каталоге новые вложенные каталоги наследуют эту запись.
  - **NO_PROPAGATE_INHERIT** – Указывает, что в этом каталоге новые вложенные каталоги наследуют эту запись, но не передают ее создаваемым в них вложенным каталогам.
  - **INHERIT_ONLY** – Указывает, что эта запись не применяется к данному каталогу, но применяется к новым объектам, которые наследуют эту запись.
  - **IDENTIFIER_GROUP** – Указывает, что значение поля **Кто** представляет собой группу. В противном случае в поле **Кто** указан пользователь или особое значение поля **Кто**.
- Поле **Кто** – Это поле содержит одно из следующих значений:
  - `Пользователь` – Указывает пользователя, к которому относится запись.
  - `Группа` – Указывает группу, к которой относится запись.
  - `Специальный` – Этот атрибут может иметь одно из следующих значений:
    - `OWNER@` – Указывает, что эта запись относится к владельцу объекта.
    - `GROUP@` – Указывает, что эта запись относится к группе, которой принадлежит объект.
    - `EVERYONE@` – Указывает, что эта запись относится ко всем пользователям системы, включая владельца и группу.

Если ACL имеет пустое значение, то доступ к объекту получает только действующий ИД пользователя 0. Для владельца объекта неявным образом установленные следующие значения маски независимо от значения ACL, даже если оно пустое:

- `READ_ACL`
- `WRITE_ACL`
- `READ_ATTRIBUTES`
- `WRITE_ATTRIBUTES`

Значение `APPEND_DATA` реализовано как `WRITE_DATA` . В действительности, функциональное различие между значением`WRITE_DATA` и значением `APPEND_DATA`. Оба значения должны устанавливаться или сбрасываться согласованно.

Владельца объекта можно изменить с помощью значения `WRITE_OWNER`. При изменении группы, которой принадлежит объект, бит **setuid** выключается. Флаги наследования имеют значение только в ACL каталога и применяются только к объектам, создаваемым в каталоге после установки флагов наследования (например, изменения в ACL родительского каталога не влияют на существующие объекты). Записи в ACL NFS v4 зависят от порядка обработки. Для того чтобы определить, разрешен ли запрошенный доступ, обрабатывается каждая запись по порядку. Рассматриваются только записи, имеющие следующие значения:

- Поле **Кто**, значение которого совпадает с действительным ИД пользователя
- Пользователь, указанный в записи или действительном ИД группы
- Группа, указанная в записи субъекта

Каждая запись обрабатывается до тех пор, пока все биты доступа субъекта, который отправил запрос, не будут иметь значение ALLOWED. После того, как запись предоставляет тип доступа ALLOWED, она больше не учитывается при обработке дальнейших записей. Если это значение маски требует доступа инициатора запроса, а права доступа не определены и при этом обнаружена запись DENY, то запрос отклоняется. Когда в процессе оценки достигается конец ACL, запрос отклоняется.

Максимальный поддерживаемый размер ACL составляет 64 Кб. Каждая запись в ACL имеет переменную длину, и 64 Кб - это единственное ограничение для записи.

Значение WRITE OWNER:

Стратегия NFS v4 обеспечивает контроль над правами чтения и записи атрибутов объекта.

Субъект с действительным ИД пользователя 0 в любом случае может переопределить стратегию NFS v4. Владелец объекта может предоставлять другим пользователям права на чтение и запись атрибутов объекта. с помощью атрибутов маски ACL `READ_ATTRIBUTES`, `WRITE_ATTRIBUTES`, `READ_NAMED_ATTRS` и `WRITE_NAME_ATTRS`. Владелец может определять права доступа на чтение и запись ACL с помощью значений маски ACL `READ_ACL` и `WRITE_ACL`. Владелец объекта всегда имеет доступ с атрибутами `READ_ATTRIBUTES`, `WRITE_ATTRIBUTES`, `READ_ACL` и `WRITE_ACL`. Владелец объекта может предоставлять другим пользователям права на изменение владельца и группы объекта с помощью атрибута `WRITE_OWNER`. Владелец объекта не может изменять владельца или группу объекта по умолчанию, но может добавить запись `WRITE_OWNER` в ACL, указав в ней себя, либо объект может наследовать запись ACL, в которой указана запись `WRITE_OWNER` со значением поля **Кто**, равным `OWNER@`. При изменении группы, которой принадлежит объект, бит **setuid** выключается.

Ниже приведены некоторые исключения из правил:

- Если объект принадлежит ИД пользователя 0, то только ИД пользователя 0 может сменить владельца, но группу может сменить субъект с атрибутом `WRITE_OWNER`.
- Если объект имеет атрибут `WRITE_OWNER` для субъекта, то в версиях AIX 5.3 до Technology Level 5300-05, если владелец объекта не имеет ИД пользователя 0, то владельца можно заменить только другим пользователем, ИД пользователя которого не равен 0. В AIX с 5300-05 и выше, если владелец объекта не имеет ИД пользователя 0, то владельца можно сменить только на действующий ИД пользователя субъекта, который пытается сменить владельца.
- Группу можно заменить любой группой из числа групп, которые параллельны группе субъекта, но ни в коем случае не на группой с ИД группы 0 или 7 (система или защита) даже если эти две группы параллельны группе субъекта.

Поддерживаемые административные базы данных на базе LDAP и на базе файлов:

Оценка не поддерживает административную базу данных NFS. Такие методы идентификации, как DCE и NIS, не поддерживаются.

Оценка поддерживает только следующее:

- Идентификация на базе файлов (по умолчанию)
- Идентификация на базе LDAP в стиле UNIX (используйте сервер LDAP IBM Tivoli Directory Server версии 6.0)

Дополнительные сведения об идентификации на базе файлов приведены в разделе Идентификация пользователей.

Идентификация LDAP:

I&A на основе LDAP настроена на режим идентификации "UNIX-type". В этом режиме административная информация (включая имена пользователей, ИД и пароли) хранится в LDAP, где доступ к данным предоставляется только администратору LDAP.

Когда пользователь входит в систему, система связывается с сервером LDAP по соединению SSL с использованием учетной записи администратора LDAP, получает необходимые данные о пользователе (включая пароль) из LDAP, а затем проводит идентификацию с помощью данных, полученных от LDAP. Система работает с административной базой данных на сервере LDAP. Остальные хосты импортируют административную информацию с этого сервера LDAP, используя описанный выше механизм. Система обрабатывает согласованной административной базой данных, внося все административные изменения на указанный сервер LDAP. ИД пользователя на любом компьютере относится к одному и тому же лицу на

всех остальных компьютерах. Кроме того, конфигурация пароля, преобразование имя-ИД и другие данные являются идентичными на всех хостах распределенной системы.

Дополнительные сведения о настройке идентификации LDAP приведены в разделе Упрощенный протокол доступа к каталогам. Дополнительные сведения о настройке SSL на LDAP приведены в разделе Настройка SSL на сервере LDAP и Настройка SSL на клиенте LDAP.

Сервер LDAP:

Команда **mksecldap -s** настраивает систему AIX как сервер LDAP для идентификации и управления данными.

Выполните следующие действия:

- Примените схему RFC2307AIX с опцией **-S**.
- Настройте сервер для использования Secure Sockets Layer (SSL) с помощью опции **-k**. Для этого требуется установить набор файлов **GSKit V8** и набор файлов **idsldap.clt_max_crypto32bit63.rte** для 32-разрядных систем или набор файлов **idsldap.clt_max_crypto64bit63.rte** для 64-разрядных систем. С помощью утилиты ikeyman создайте пары ключей для сервера каталогов.

Опция пользователя LDAP должны отвечать требованиям, предъявляемым при оценке. Схема RFC2370AIX определяет атрибуты пользователя. Используйте те же значения, которые описаны в разделе Настройка системы BAS/EAL4+. Администраторы Tivoli Directory Server не обязаны периодически изменять свои пароли (например, для паролей администраторов отсутствует значение **MaxAge**). По этой причине пароль администратора LDAP должен изменяться так же часто, как и пароль пользователя AIX (**MaxAge** = 8 (в неделях)).

В Tivoli Directory Server 6.3 обработка сбоев при идентификации не применяется к администратору каталога и к членам группы администраторов. Правила составления пароля также не применяются к учетным записям администратора. Эти правила необходимо применять принудительно, если используется Tivoli Directory Server 6.3.

Если администратор не использует общую базовую программу базы данных LDAP для управления пользователями, то он должен обеспечить согласованность управления базой данных, которая содержит идентификационные данные пользователя, в различных частях системы TCP Offload Engine (TOE) в одной сети. Приведем примеры:

- `/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`

Информация, связанная с данной:

Сведения о пакетах, наборах файлов и предварительных требованиях сервера каталогов IBM Tivoli Directory Server

Клиент LDAP:

Команда **mksecldap -c** настраивает систему AIX как клиента LDAP для идентификации и управления данными.

Выполните следующие действия:

- С помощью команды **mksecldap -c** укажите **unix_auth** для **authType** с опцией **-A**.
- Настройте клиент на использование SSL с помощью опции **-k** команды **mksecldap -c**. Для того чтобы указать ключ SSL клиента, требуется установить набор файлов **GSKit** и **ldap.max_crypto_client**. Для создания пар ключей для сервера каталогов используйте служебный класс **gsk7ikm**.

NFS v4 Client/Server и Kerberos:

Среда NFS v4 Client/Server включает в себя LDAP для работы с идентификационными данными и Kerberos для установления защищенного канала между клиентами и серверами NFS v4. Оцениваемая конфигурация поддерживает NAS v1.4 для Kerberos и сервер каталогов IBM Tivoli Directory Server v6.0 (сервер LDAP) для базы данных пользователей.

NAS v1.4 (Сервер Kerberos версии 5) должен быть настроен на использование LDAP для базы данных. Паспорта Kerberos, предварительно выданные сервером Kerberos действуют до их устаревания.

Пр использовании идентификации Kerberos одноразовое разрешение, использующееся в вызовах удаленных процедур пользователем, связывается с текущим паспортом Kerberos, принадлежащим пользователю, и на него не влияет настоящий или действующий ИД пользователя процесса. При попытке доступа к удаленной файловой системе NFS с помощью идентификации удаленный при работе программы **setuid** ИД пользователя, который виден на сервере, основывается на субъекте Kerberos, а не на ИД пользователя, который является владельцем выполняющейся программы **setuid**.

Конфигурация требует настройки NFS на использование защиты RPCSEC-GSS. Дополнительная информация приведена в разделах Сетевая файловая система, Настройка сервера NFS иНастройка клиента NFS. При настройке сервера выберите идентификацию Kerberos и включите усиленную защиту на сервере. Включить ее можно с помощью SMIT с использованием команды **chnfs**. Команда **chnfs** имеет опцию включения защиты RPCSEC_GSS. При настройке клиента следуйте указаниям по использованию Kerberos в разделеНастройка клиента NFS. Указания по настройке сервера данных Kerberos с использованием шифрования DES3 для защиты приведены в разделе Настройка сети для RPCSEC-GSS. Конфигурация поддерживает только шифрование des3.

Правила для паролей:

При использовании сервера Kerberos с LDAP в качестве базы данных в конфигурации следует использовать следующие значения для правил паролей.

Дополнительные сведения о правилах для паролей приведены в разделе "Раздел 9. Управление паролями Службы сетевой идентификации" в *Руководстве по службе сетевой идентификации IBM версии 1.4 для администраторов и пользователей AIX, Linux и Solaris*.

Список значений:

mindiff
4

maxrepeats
2

minalpha
2

minother
2

minlen 8

minage
0

histsize
10

Для безопасной связи между клиентом AIX NFS v4 и сервером AIX NFS v4 с явным использованием только типов кодирования DES3 создайте субъект сервера "nfs/hostname" с типом кодирования DES3 (например, des3-cbc-sha1), а также соответствующую запись в файле `keytab` (с помощью интерфейса **kadmin**) и сделайте DES3 (например **des3-cbc-sha1**) первой записью в разделе **default_tgs_enctypes** файла `/etc/krb5/krb5.conf` в системе, которая является клиентом NFS v4.

Сервер виртуального ввода-вывода:

Сервер виртуального ввода-вывода (VIOS) расположен в отдельном разделе LPAR. Он обеспечивает базовый самостоятельный контроль доступа между драйверами устройств VIOS SCSI, действуя от имени разделов LPAR и логических и физических томов на основе SCSI посредством преобразований.

Раздел LPAR (с помощью драйвера устройства VIOS SCSI) может быть преобразован в 0 и более логических и физических томов, но том может быть преобразован только в один раздел LPAR. Это преобразование ограничивает раздел LPAR только теми томами, которые для него назначены. Также VIOS контролирует преобразование драйверов устройства адаптера Ethernet VIOS в драйверы устройств Ethernet VIOS, действуя от имени групп разделов LPAR, которые совместно используют виртуальную сеть. В оцениваемой конфигурации допускается только преобразование драйвера устройства адаптера Ethernet в драйвер устройства Ethernet по типу один к одному, выполняемое от имени группы разделов LPAR. Преобразование по типу один к одному настраивается администратором и применяется драйверами устройств. Кроме того, пакеты Ethernet не должны снабжаться тэгом VLAN в оцениваемой конфигурации. Этот механизм можно использовать для ограничения разделов LPAR, которым видны определенные пакеты Ethernet.

Интерфейс VIOS должен быть защищен от доступа пользователей, не имеющих прав доступа. Опции пользователя VIOS должны отвечать требованиям, предъявляемым при оценке. Фактическим требованием является то, что TSF должна предоставлять механизм проверки удовлетворения шифрами следующего показателя качества: вероятность того, что шифр может быть получен атакующим во время срока действия шифра, должна быть меньше, чем 2^-20. Для пользователя в каталоге `/etc/security/user` должны быть изменены следующие параметры:

maxage
8

maxexpired
1

minother
2

minlen 8

maxrepeats
2

loginretries
3

histexpire
52

histsize
20

Для изменения значений, установленных по умолчанию, используются следующие команды:

```
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
```

Когда главный администратор (**padmin**) создает нового пользователя, атрибуты этого пользователя должны быть указаны явно. Например, для создания пользователя с именем пользователя *davis* **padmin** должен использовать следующую команду:

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

Также **padmin** должен остановить следующие демоны и затем перезагрузить систему:

- Для удаления **writesrv** и **ctrmc** из файла `/etc/inittab`:

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

- Для того чтобы демон не запускался при загрузке, удалите файлы `/etc/rc.d/rc2.d/Ksshd` и `/etc/rc.d/rc2.d/Ssshd`. После перезагрузки остановите демоны RSCT:

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

Независимо от ролей все пользователи должны считаться пользователями с правами администратора.

Системный администратор может запускать все команды, кроме тех, которые перечислены в следующем списке и могут использоваться только главным администратором (**padmin**):

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

Управление входом в систему

После установки системы можно изменить параметры экрана входа в систему для обеспечения безопасности системы.

Даже просмотр приглашения на вход в систему AIX может дать злоумышленникам ценную информацию - например, имя хоста и версию операционной системы. Эти сведения могут пригодиться им при выборе способов проникновения в систему. По соображениям защиты рекомендуем вам изменить параметры входа в систему как можно скорее после первоначальной установки системы.

Для рабочих столов KDE и GNOME характерно много общего в вопросах защиты. Дополнительная информация о KDE и GNOME приведена в публикации *Установка и миграция*.

Сведения о пользователях, группах и паролях приведены в разделе “Пользователи, группы и пароли” на стр. 47.

Настройка входа в систему:

Параметры входа в систему можно настроить в файле `/etc/security/login.cfg`.

Для того чтобы максимально затруднить проникновение в систему путем угадывания пароля настройте управление входом в систему в файле `/etc/security/login.cfg` в следующим образом:

Таблица 1. Атрибуты и рекомендуемые значения управления входом в систему.

Атрибут	Относится к PtY (сетевые устройства)	Относится к TTY	Рекоменду- емое значение	Комментарий
sak_enabled	Да	Да	false	Ключ Secure Attention нужен довольно редко. Дополнительная информация приведена в разделе “Работа с защищенной клавишей внимания” на стр. 5.
logintimes	Нет	Да		Максимальное разрешенное количество попыток входа в систему.
logindisable	Нет	Да	4	Возможность входа в систему с конкретного терминала будет запрещена после 4 неудачных попыток.
logininterval	Нет	Да	60	Терминал будет отключен, если указанное число неудачных попыток входа в систему будет выполнено в течение 60 секунд.
loginreenable	Нет	Да	30	Терминал будет вновь включен через 30 минут после автоматического отключения.
logindelay	Да	Да	5	Задержка между выдачей приглашений на вход в систему. Это значение будет увеличиваться на его начальное значение после каждой неудачной попытки входа в систему (например; если начальное значение - 5 секунд, будут последовательно применяться значения 5, 10, 15 и 20).

Учтите что следующие ограничения, установленные для портов, главным образом влияют на локальные терминалы, подключенные к последовательным портам, и практически не влияют на сетевые псевдотерминалы. В этом файле можно задать индивидуальные ограничения для отдельных терминалов, например:

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

Изменение приветствия при входе в систему:

Изменив значение параметра *herald* в файле /etc/security/login.cfg, можно запретить выдачу определенных сведений в приглашении на вход в систему.

Параметр *herald* содержит стандартное приглашение на вход в систему. Значение этого параметра можно изменить с помощью команды **chsec** или вручную.

Ниже приведен пример команды **chsec** для изменения значения по умолчанию параметра *herald*:

```
# chsec -f /etc/security/login.cfg -s default
-a herald="Доступ неуполномоченных пользователей к этой системе запрещен.\n\nlogin:"
```

Дополнительные сведения о команде **chsec** приведены в разделе *Справочник по командам, том 1*.

Если вы хотите отредактировать файл `/etc/security/login.cfg` вручную, откройте его в любом редакторе и укажите для параметра *herald* следующий текст:

```
default:
herald ="Доступ неуполномоченных пользователей в систему запрещен.\n\nlogin:"
 sak_enable = false
 logintimes =
 logindisable = 0
 logininterval = 0
 loginreenable = 0
 logindelay = 0
```

Примечание: Для повышения защищенности системы присвойте переменным *logindisable* и *logindelay* значения больше `0` (`# > 0`).

Изменение приветствия при входе в систему:

Этот раздел относится к пользователям Общая среда рабочего стола (CDE). Стандартное приветствие, выдаваемое при входе в CDE, содержит имя хоста и версию операционной системы. Для того чтобы эта информация не отображалась, откройте файл `/usr/dt/config/$LANG/Xresources`, где **$LANG** - локальный язык вашей системы.

Например, если для **$LANG** указано значение **C**, то скопируйте этот файл в каталог `/etc/dt/config/C/Xresources`. Затем откройте файл `/usr/dt/config/C/Xresources` в редакторе и удалите из него сведения об имени хоста и версии операционной системы.

Дополнительные рекомендации по защите, связанные с CDE, приведены в разделе “Рекомендации по работе с X11 и CDE” на стр. 40.

Запрет отображения имени пользователя и изменение приглашения ввода пароля:

В защищенной среде может возникнуть необходимость запретить отображение имени пользователя при входе в систему или указать приглашение ввода пароля, отличное от стандартного.

Ниже показано, каким образом выглядит приглашение входа в систему по умолчанию:

```
Имя пользователя: foo
Пароль пользователя foo:
```

Для того чтобы запретить отображение имени пользователя в приглашениях и сообщениях об ошибках, измените параметр *usernameecho* в файле `/etc/security/login.cfg`. По умолчанию для параметра *usernameecho* указано значение 'true', т.е. имя пользователя разрешено показывать. Значение этого параметра можно изменить с помощью команды **chsec** или вручную.

Ниже приведен пример применения команды **chsec** для изменения значения по умолчанию параметра *usernameecho*:

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

Дополнительные сведения о команде **chsec** приведены в разделе *Справочник по командам, том 1*.

Если вы хотите внести изменения в файл `/etc/security/login.cfg` вручную, откройте его в любом редакторе и укажите следующий текст для параметра *usernameecho*:

```
default:
 usernamecho = false
```

Значение 'false' параметра *usernameecho* указывает, что имя пользователя в приглашении входа в систему показывать запрещено. Вместо имени пользователя в приглашениях и сообщениях об ошибках отображаются символы '*', как это показано ниже:

```
Имя пользователя:
Пароль пользователя ***:
```

При необходимости вместо стандартного приглашения ввода пароля вы можете указать собственное. Для этого применяется параметр *pwdprompt* в файле `/etc/security/login.cfg`. По умолчанию отображается строка "Пароль пользователя *имя-пользователя*:", где *имя-пользователя* заменяется именем пользователя.

Значение этого параметра можно изменить с помощью команды **chsec** или вручную.

Ниже приведен пример применения команды **chsec** для изменения значения по умолчанию параметра *pwdprompt* на значение "Пароль:":

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

Если вы хотите внести изменения в файл `/etc/security/login.cfg` вручную, откройте его в любом редакторе и укажите следующий текст для параметра *pwdprompt*:

```
default:
 pwdprompt = "Пароль: "
```

В результате изменения параметра *pwdprompt* строка "Пароль: " будет отображаться в приглашении входа в систему, а также другими приложениями, в которых применяется приглашение ввода пароля. После настройки приглашение входа в систему будет выглядеть следующим образом:

```
Имя пользователя: foo
Пароль:
```

Настройка параметров входа в систему по умолчанию:

Параметры входа в систему по умолчанию задаются в файле `/etc/security/login.cfg`.

Для настройки базовых значений по умолчанию для различных параметров входа в систему (число попыток входа в систему, разрешение на вход в систему после запрета и т.д.), которые могут применяться, например, при создании новых пользователей, можно отредактировать файл `/etc/security/login.cfg`.

Защита терминалов, оставленных без внимания:

Для защиты терминала служат команды **lock** и **xlock**.

Любая система окажется незащищенной, если к ней будут подключены терминалы, в которых пользователи вошли в систему и физически отошли от терминала. Самая большая опасность - это оставление своего рабочего места администратором системы, если при этом у него запущен терминал с правами доступа root. Пользователи должны выходить из системы всегда, когда они физически отходят от своих терминалов. Оставление терминала без внимания - это серьезное нарушение безопасности. Терминал можно блокировать с помощью команды **lock**. Интерфейс AIXwindows можно заблокировать командой **xlock**.

Включение автоматического выхода из системы:

Принудительное отключение от системы позволяет предотвратить нарушение защиты системы злоумышленником.

Еще одно распространенное нарушение безопасности заключается в том, что пользователи не выходят из системы в течение длительного времени. Это увеличивает вероятность того, что злоумышленник сможет получить доступ к терминалу пользователя, что может привести к серьезным последствиям.

Для того чтобы минимизировать этот риск, можно задействовать режим принудительного отключения пользователей от системы. Для этого задайте переменным среды TMOUT и TIMEOUT число секунд неактивности. По истечении времени неактивности автоматически будет выполнен выход из системы, как в следующем примере:

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

В этом примере задано 600 секунд, что равно 10 минутам. Этот метод работает только из приложения-оболочки. Для защиты переменных от переопределения их можно сделать неизменяемыми следующим образом:

```
readonly TMOUT TIMEOUT
```

Переменные среды TMOUT и TIMEOUT определяются в файлах .profile пользователей или в файле `/etc/security/.profile`. Это позволяет добавить файл в файл .profile при создании пользователя.

Защита с помощью отключения работы со стеком

Защита компьютерных систем - важный аспект бизнеса "по требованию". В мире современных сетевых сред чрезвычайно важно защититься от разнообразных атак.

В настоящее время атаки на компьютерные системы становятся все изощреннее, и вследствие этого растет вероятность нарушения деятельности коммерческих и государственных компаний. Так как никакие защитные меры не могут гарантировать всесторонней защиты от атак, то необходимо предотвращать нападения, используя несколько технологий одновременно. В этом разделе описывается технология защиты, используемая в AIX для предотвращения атак, направленных на переполнение буфера.

Пробить защиту можно многими способами, но самый распространенный из них - это наблюдение за системными средствами администрирования, поиск и использование в своих целях переполнения буфера. Атаки переполнения буфера происходят во время перезаписи внутреннего программного буфера, поскольку в этот момент невозможна строгая проверка данных(например, командная строка, переменные среды, операции ввода-вывода для диска или терминала). Вредоносный код через переполнение буфера встраивается в работающий процесс и изменяет схему его выполнения. Адрес возврата функции перезаписывается и перенаправляется в место вставленного кода. Основные последствия таких атак - это проверка неправильных или несуществующих границ и искаженная информация о действительных источниках данных. Например, переполнение буфера может возникнуть, если размер объекта данных рассчитан на хранение 1 Кб данных, а программа, не проверив границы ввода, может скопировать в этот объект больше, чем 1 Кб данных.

Целью злоумышленника является атака на команды и/или утилиты, предоставляющие обычному пользователю права доступа root. Имея все права, можно получить контроль над программой и разрешить переполнение буфера. Обычно атаки нацелены на установку ИД пользователя root (UID) или на программы, запускающие командный процессор и таким образом позволяющие получить доступ к системе с правами root.

Такие атаки можно предотвратить, блокируя выполнение вредоносного кода, введенного при переполнении буфера. Запретите выполнение процессов в тех областях памяти, в которых вообще ничего не должно выполняться (области стека и кучи).

Механизм защиты буфера SED от переполнения:

В AIX предусмотрен механизм отключения обработки стека (SED), предназначенный для того, чтобы обработка шла не в стеке/, а в областях данных.

Благодаря отключению и прерыванию работы несанкционированной программы злоумышленник лишается возможности получить права пользователя root с помощью атаки переполнения буфера. Несмотря на то, что эта функция не прекращает переполнение буфера, она дает защиту благодаря отключению атак на переполненные буферы.

Начиная с процессоров семейства POWER4, появилась возможность выполнения активизации и/или деактивизации для памяти на уровне страницы. Механизм AIX SED использует эту встроенную аппаратную поддержку для реализации функции "невыполнения" в выборочных областях памяти. Если эта функция активна, операционная система проверяет и помечает различные файлы во время работы исполняемой программы. Затем диспетчеру памяти операционной системы и диспетчеру процессов отправляется предупреждение о том, что для создаваемого процесса активна функция SED. Выбранные области памяти помечаются для "невыполнения". Если в какой-либо из помеченных областей возникает исключительная ситуация, то на аппаратном уровне взводится флаг исключения, и ОС останавливает соответствующий процесс. Сведения об исключительной ситуации и приложении заносятся в протокол ошибок AIX.

Механизм SED реализован главным образом командой **sedmgr**. Команда **sedmgr** позволяет управлять системным режимом работы SED и настраивать исполняемые файлы с помощью флагов SED.

Режимы и мониторинг SED:

Механизм отключения работы со стеком (SED) в AIX реализован с помощью системных флагов режима и отдельных флагов файловых заголовков.

Системные флаги управляют системными операциями SED, а флаги на уровне файла указывают, как файлы должны обрабатываться в SED. Механизм защиты от переполнения буфера (BOP) предусматривает четыре режима работы:

off Механизм SED отключен, ни один процесс для защиты с помощью SED не помечается.

select Защита SED работает только для выбранного набора файлов. Выбранный набор файлов определяется путем просмотра связанных с SED флагов в двоичном заголовке исполняемой программы. Заголовок исполняемой программы позволяет флагам, связанным с SED, запросить включение в режим **select**.

setidfiles
Позволяет включить SED не только для файлов, запросивших этот механизм, но и для всех важных системных файлов **setuid** и **setgid**. В этом режиме операционная система обеспечивает SED не только для файлов с установленным SED-флагом **request**, но и для исполняемых файлов со следующими характеристиками (кроме файлов, в заголовках которых стоит флаг *exempt*):

- файлы SETUID, владельцем которых является root
- Файлы SETGID, основной для которых является группа **system** или **security**

все Все исполняемые программы, загруженные в систему, защищаются с помощью SED, за исключением файлов, запросивших освобождение от SED. Флаги освобождения входят в заголовок исполняемой программы.

Функция SED в AIX также предоставляет возможность не остановки, а отслеживания процесса при возникновении исключительной ситуации. Такой метод системного контроля позволяет системному администратору выявлять точки прерывания и ошибки в системной среде перед развертыванием SED в готовой системе.

В команде **sedmgr** есть параметр для включения SED, чтобы при возникновении исключительной ситуации не процесс останавливался, а только отслеживались файлы. Системный администратор может определить, будет ли работать программа при нормальной работе со стеком. Этот параметр работает в паре с системным режимом с параметром `-c`. Если включен режим **монитора**, то система продолжит процесс даже при возникновении исключительной ситуации, связанной с SED. Вместо останова процесса операционная система занесет исключительную ситуацию в протокол ошибок AIX. Если мониторинг SED выключен, то любой процесс, вызвавший исключительную ситуацию, связанную с SED, будет остановлен операционной системой.

Для вступления в силу каких-либо изменений системных флагов режима SED требуется перезагрузка системы. Все события такого типа контролируются.

Флаги SED для исполняемых программ:

В AIX можно пометить исполняемые файлы механизма SED с помощью команды **sedmgr**.

Для поддержки двух новых флагов SED, позволяющих включить в заголовки исполняемых программ параметры `select` и `exempt`, был расширен компоновщик. Флаг **select** позволяет исполняемому файлу запрашивать и получать защиту SED в режиме `select` системной операции SED, а флаг **exempt** позволяет исполняемому файлу запросить освобождение от защиты SED. Такие исполняемые файлы могут выполняться в любой области памяти процесса.

Флаг освобождения позволяет системному администратору наблюдать за работой механизма SED и правильно оценивать ситуацию. При необходимости работу приложения в областях стека и данных можно разрешить, но при этом необходимо помнить о потенциальной опасности.

Влияние системных и файловых параметров на режим SED проиллюстрировано следующей таблицей:

Таблица 2. Влияние системных и файловых параметров на режим SED

	Флаги SED для исполняемых файлов			
Системный режим SED	**request**	**exempt**	**system**	**Файлы Setuid-root или setgid-system/security**
off	–	–	–	–
select	enabled	–	–	–
setgidfiles	enabled	–	–	enabled
все	enabled	–	enabled	enabled

Вопросы и замечания по SED:

По умолчанию функция AIX SED находится в режиме **select**. Программы **setuid** и **setgid** поддерживают режим **select** и по умолчанию работают в защищенном режиме.

Активизация механизма SED может привести к повреждению более старых бинарных файлов, если они не поддерживают обработку функции "невыполнения" в областях кучи стека. Такие приложения следует выполнять в областях данных стека. Системный администратор в такой ситуации может пометить файл для освобождения с помощью команды **bopmgr**. В AIX Java™ 1.3.1 и AIX Java 1.4.2 предусмотрены компиляторы Just-In-Time (JIT), которые динамически генерируют и выполняют встроенный код объекта во время работы приложений на Java (Решение о компилируемом коде принимается виртуальной машиной Java на основе профайла выполнения приложения). Этот код объекта хранится в буферах данных, выделяемых компилятором JIT. Поэтому, если AIX настроен для работы в режиме SED **ALL**, то системный администратор должен установить флаг освобождения бинарного файла Java.

Если флаги, связанные с SED, изменяются для исполняемого файла, то они будут применены при следующей загрузке и выполнении этого файла. На работающие процессы изменения не влияют. Функция SED контролирует и отслеживает и 32-, и 64-разрядные программы как на уровне системы, так и на уровне файлов. Эта функция доступна только когда ОС AIX работает с 64-разрядным ядром.

Связанная информация

Описание команды **sedmgr**

Утилита ведения протокола ошибок AIX

Рекомендации по работе с X11 и CDE

В этом разделе обсуждаются слабые места системы защиты, которые могут быть вызваны применением сервера X X11 и общей среды рабочего стола (CDE).

Удаление файла /etc/rc.dt:

В системах, для которых требуется высокий уровень защиты, следует удалить файл `/etc/rc.dt`.

Хотя графический интерфейс CDE удобен для работы, его применение в некоторых аспектах снижает защиту системы. По этой причине не следует пользоваться CDE на серверах, для которых должен быть обеспечен высокий уровень защиты. Рекомендуется вообще не устанавливать наборы файлов CDE (dt). Если вы уже установили эти наборы файлов, рекомендуем вам удалить их, и в первую очередь это касается сценария `/etc/rc.dt`, запускающего CDE.

Дополнительная информация о CDE приведена в разделе *Управление операционной системой и устройствами*.

Предотвращение удаленного слежения за сервером X:

Очень серьезная угроза для защиты сервера заключается в том, что в системе предусмотрена возможность удаленного мониторинга сервера X11.

С помощью команд **xwd** и **xwud** можно следить за работой сервера X и перехватывать данные, вводимые с клавиатуры, что потенциально может привести к рассекречиванию паролей и другой конфиденциальной информации. Во избежание возникновения подобных проблем следует удалить эти выполняемые файлы, если только они не нужны для выполнения каких-либо важных задач, либо как минимум запретить доступ к этим файлам для всех пользователей, кроме root.

Команды **xwd** и **xwud** входят в набор файлов`X11.apps.clients`.

Если в вашей среде обязательно должны применяться команды **xwd** и **xwud**, рекомендуется воспользоваться приложениями OpenSSH или MIT Magic Cookies. Эти программы устраняют факторы риска для системы защиты, вносимые командами **xwd** и **xwud** .

Дополнительная информация по программам OpenSSH и MIT Magic Cookies приведена в документации по ним.

Включение и отключение управления доступом:

Сервер X позволяет открывать на локальном рабочем столе окна с удаленных компьютеров. Для того чтобы разрешить удаленный доступ, нужно выполнить команду **xhost +**.

Обязательно укажите имя конкретного хоста в команде **xhost +**, так как в противном случае доступ к серверу X будет разрешен всем удаленным хостам. Если в этой команде будет указано имя хоста, то доступ к серверу будет предоставлен только указанному хосту. Команду **xhost** нужно вызывать в следующем формате:

```
# xhost + hostname
```

Если вы не укажете имя хоста, то доступ будет предоставлен всем хостам.

Дополнительные сведения о команде **xhost** приведены в разделе *Справочник по командам*.

Запрет выполнения команды xhost:

Несанкционированное выполнение команды **xhost** можно предотвратить с помощью команды **chmod**.

По соображениям защиты рекомендуется запретить доступ к команде **xhost** всем пользователям, кроме root. Для этого следует установить режим доступа к файлу `/usr/bin/X11/xhost` равным 744 с помощью команды **chmod**:

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

Список программ setuid/setgid

В системе AIX существуют различные программы setuid/setgid. Эти права можно удалить для программ, которые не должны быть доступны обычным пользователям.

В пакет обычной установки AIX включены следующие программы. В системе AIX, настроенной CC, этот список сокращен и включает меньше программ.

- /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

Пользователи, группы и пароли

В этом разделе описаны различные способы управления пользователями и группами AIX.

Автоматическое создание домашнего каталога при входе в систему

Операционная система AIX может автоматически создавать домашний каталог во время входа пользователя в систему.

Эта функция полезна для удаленных пользователей (например, пользователей, определенных на сервере LDAP), которые могут не иметь домашний каталог в локальной системе. Операционная система AIX обладает двумя механизмами автоматического создания домашних каталогов пользователей при входе: обычный механизм AIX и механизм на основе PAM. Оба механизма могут работать одновременно.

Механизм AIX

Механизм AIX работает с командами **getty**, **login**, **rlogin**, **rsh**, **telnet** и **tsm**. Механизм AIX поддерживает идентификацию STD_AUTH и PAM_AUTH с помощью модуля pam_aix. Механизм AIX включается установкой атрибута mkhomeatlogin в разделе "usw" файла **/etc/security/login.cfg** в значение `true` (см. файл `/etc/security/login.cfg`). Функция automatic-home-directory-creation-at-login включается и выключается командой **chsec**. Например, чтобы включить эту функцию, введите в консоли:

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

Когда эта функция включена, процесс обработки входа в систему после завершения идентификации пользователя проверяет наличие у него домашнего каталога. Если каталог не существует, процесс его создает.

Примечание: Атрибут **`mkhomeatlogin`** поддерживается только в AIX 5L версии 6.1 с технологическим пакетом обслуживания 6100-02 и более поздних.

Механизм PAM

В состав системы AIX входит модуль pam_mkuserhome для создания домашних каталогов при использовании механизма PAM. Модуль pam_mkuserhome может быть помещен в стек вместе с другими модулями сеанса для служб входа в систему. Для активации модуля в службу необходимо добавить запись. Например, чтобы включить создание домашнего каталога при входе в систему с помощью команды **telnet**, использующей PAM; добавьте в файл **/etc/pam.cfg** следующую запись:

```
telnet session optional pam_mkuserhome
```

ИД учетной записи

Каждая учетная запись пользователя снабжается уникальным числовым ИД. Операционная система AIX предоставляет права доступа на основании ИД учетной записи.

Очень важно знать, что учетные записи с одинаковыми ИД виртуально представляют собой одну и ту же учетную запись. При создании пользователей и групп команды **mkuser** и **mkgroup** системы AIX всегда проверяют целевой реестр, чтобы избежать конфликтов с ИД уже существующих учетных записей.

Также систему можно настроить так, чтобы при создании учетной записи все реестры пользователей (групп) проверялись с помощью системного атрибута **dist_uniqid**. Атрибут **dist_uniqid** раздела usw в файле `/etc/security/login.cfg` можно изменять с помощью команды **chsec**. Для настройки проверки конфликтов ИД по всем реестрам выполните команду:

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

Для атрибута **dist_uniqid** существует три допустимых значения:

never Это значение запрещает проверку конфликтов ИД по нецелевым реестрам (значение по умолчанию).

always Это значение включает проверку конфликтов ИД по всем реестрам. При обнаружении конфликта между целевым и каким-либо другим реестром команда **mkuser** (**mkgroup**) берет уникальный ИД, которого нет ни в одном реестре. Не выполняется это только в том случае, если значение ИД указано в командной строке (например, `mkuser id=234 foo`, а ИД 234 уже присвоен пользователю в каком-либо реестре).

uniqbyname

Это значение включает проверку конфликтов ИД по всем реестрам. Конфликты между реестрами допускаются только если учетная запись имеет тот же ИД, что и существующая учетная запись для команд типа `mkuser id=123 foo`. Если ИД задается не из командной строки, то ИД новой учетной записи может совпадать или не совпадать с ИД существующей учетной записи с тем же именем в

другом реестре. Например, есть локальная учетная запись *acct1* с ИД 234. При создании учетной записи LDAP *acct1* команда `mkuser -R LDAP acct1` может присвоить ей уникальный ИД 235. В результате получаем локальную учетную запись *acct1* с ИД 234 и учетную запись LDAP *acct1* с ИД 235.

Примечание: Проверка конфликтов ИД в целевом реестре выполняется всегда, независимо от значения атрибута **dist_uniqid**.

Значение **uniqbyname** хорошо работает с двумя реестрами. Если реестров больше, и в каких-либо двух из них найден конфликт ИД, то при создании новой учетной записи в третьем реестре со значениями конфликтующих ИД команда **mkuser** (**mkgroup**) может работать непредсказуемо. Новая учетная запись может быть создана успешно, или не создана совсем. Это зависит от очередности проверки реестров.

Например: Есть система с тремя реестрами: локальным, LDAP и DCE. В реестре LDAP есть учетная запись *acct1*, а в реестре DCE - учетная запись *acct2*, обе с одинаковыми ИД 234. Когда системный администратор создает локальную учетную запись **uniqbyname** с помощью команды `mkuser -R files id=234 acct1` (`mkgroup -R files id=234 acct1`), в первую очередь проверяется реестр LDAP и выясняется, что ИД 234 присвоен учетной записи LDAP *acct1*. Так как новая учетная запись создается с тем же именем, команда **mkuser** (**mkgroup**) успешно создаст локальную учетную запись *acct1* с ИД 234. Если же первым будет проверяться реестр DCE, то команда **mkuser** (**mkgroup**) обнаружит, что ИД 234 присвоен учетной записи DCE *acct2*, и учетная запись *acct1* создана не будет. Проверка конфликтов ИД обеспечивает уникальность ИД между локальным и удаленными реестрами или между только удаленными реестрами. Уникальность ИД между вновь созданной учетной записью в удаленном реестре и существующими локальными пользователями других систем, работающими с тем же удаленным реестром, не гарантируется. Удаленный реестр, недоступный во время выполнения команды **mkuser** (**mkgroup**), пропускается.

Учетная запись root

Учетная запись root обладает практически неограниченным доступом ко всем программам, файлам и ресурсам системы.

Пользователь root - это особый пользователь в файле `/etc/passwd` с идентификатором (UID), равным `0`. Как правило, этому пользователю присваивается имя *root*. Таким образом, особая роль учетной записи root объясняется не именем пользователя, а значением UID, равным `0`. Это означает, что любой пользователь с нулевым значением UID обладает теми же правами доступа, что и администратор. Кроме того, учетная запись root всегда идентифицируется с помощью локальных файлов защиты.

У учетной записи root всегда должен быть пароль, и этот пароль не должен разглашаться. Пароль для учетной записи root следует задать сразу после установки системы. Пароль записи root должен быть известен только системному администратору. Системные администраторы должны выполнять от имени пользователя root только те функции управления, для которых необходимы права доступа root. Для выполнения всех остальных операций им следует переключаться на свою обычную учетную запись.

Внимание: Постоянно работая под именем root, можно повредить конфигурацию системы, так на эту учетную запись не распространяются многие ограничения в системе.

Отключение прямого доступа к учетной записи root:

Один из распространенных методов атак потенциальных взломщиков - получение пароля администратора, или пользователя root.

Для того чтобы защитить систему от таких атак, можно запретить прямой доступ к ИД root. При этом системные администраторы будут получать необходимые права доступа с помощью команды **su -**. Отключение прямого доступа к учетной записи root позволяет не только исключить возможность атаки пользователя root, но и отслеживать, кто и когда получал права доступа администратора. Для этого можно просмотреть файл `/var/adm/sulog`. Другой способ - включить функции контроля системы, которые будут отправлять уведомления о таких действиях.

Для отключения прямого доступа к учетной записи root внесите изменения в файл /etc/security/user. Укажите в параметре rlogin в записи пользователя root значение false.

Перед тем как отключить прямой доступ к учетной записи root, проанализируйте возможные ситуации, в которых системный администратор не сможет войти в систему под другим идентификатором пользователя. Например, при переполнении домашней файловой системы пользователь не может войти в систему. Если прямой доступ к учетной записи root отключен, а домашняя файловая система пользователя, который мог с помощью команды **su -** переключиться на пользователя root, переполнена, то доступ к учетной записи root может быть навсегда утерян. Во избежание таких ситуаций, системные администраторы могут создать для себя файловые системы большего размера, чем для обычных пользователей.

Учетные записи пользователей

В этом разделе описывается административные задачи защиты пользовательских учетных записей.

Рекомендуемые атрибуты пользователей:

В процессе администрирования пользователей вы создаете пользователей и группы, определяя их атрибуты.

Основным атрибутом пользователя являются его права доступа. Пользователи - это основные действующие лица в системе. С помощью атрибутов можно задать права доступа, среду, способ идентификации и параметры входа в систему.

Группы - это наборы пользователей с одинаковыми правами доступа к защищенным ресурсам. Каждой группе присвоен идентификатор; в нее могут входить обычные пользователи и администраторы. Первым администратором группы обычно является ее создатель.

Для каждой учетной записи пользователя можно задать множество атрибутов, включая пароль и параметры входа в систему. Список настраиваемых атрибутов приведен в “Система дисковых квот - обзор” на стр. 76. При задании атрибутов соблюдайте следующие правила:

- ИД пользователей не должны совпадать. Все средства защиты и учета работают только в том случае, если каждому пользователю присвоен уникальный ИД.
- Присваивайте пользователям осмысленные имена. Рекомендуется использовать их настоящие имена, поскольку в большинстве систем обработки электронной корреспонденции применяются именно ИД пользователей.
- Добавлять, изменять и удалять пользователей следует только с помощью SMIT. Хотя все эти задачи можно выполнить из командной строки, интерфейс SMIT поможет сократить число мелких ошибок.
- Не присваивайте пользователям пароли заранее. Это следует делать лишь в тот момент, когда пользователь готов к началу работы с системой. Если в поле пароля в файле /etc/passwd указана звездочка (*), то никто не сможет воспользоваться этой учетной записью для входа в систему.
- Не изменяйте системные ИД пользователей - они нужны для правильной работы системы. Эти ИД перечислены в файле /etc/passwd.
- Вообще, параметр *admin* пользователей не должен принимать значение true. Атрибуты пользователей, для которых в файле /etc/security/user указано admin=true, может изменять только пользователь учетной записи root.

Операционная система поддерживает стандартные атрибуты пользователей, которые обычно указываются в файлах /etc/passwd и /etc/system/group, например:

Информация об идентификации
Задает пароль.

Удостоверение
Задает идентификатор пользователя, а также основной и дополнительной группы.

Environment
Задает домашнюю среду или оболочку.

Максимальная длина имен пользователей и групп:

Можно настроить и получить максимальную длину имен пользователей и групп.

Длина имен пользователей и групп по умолчанию не должна превышать 9 символов. В версии AIX 5.3 и более поздней максимальное значение длины имени пользователей и групп можно увеличить с 9 до 256 символов. Так как в длине имен пользователей и групп учитывается последний символ NULL, фактическая длина составляет не более 8 или 255 символов.

Длина имен пользователей и групп задается в системном параметре **v_max_logname** для устройства sys0. Значение параметра **v_max_logname** можно запросить у ядра или в базе данных ODM, и его можно изменить. При работе системы используется значение, заданное в ядре. Значение, заданное в базе данных ODM, применяется при следующей загрузке системы.

Примечание: Если уменьшить максимальную длину имен пользователей и групп после того, как она была увеличена, то поведение системы может быть непредсказуемым. В этом случае в системе могут остаться ранее созданные длинные имена пользователей и групп.

Извлечение ограничения на длину имен пользователей и групп из базы данных ODM:

Команды и процедуры позволяют получить параметр **v_max_logname**.

С помощью команды **lsattr** можно извлечь параметр **v_max_logname** из базы данных ODM. Команда **lsattr** отображает параметр **v_max_logname** как атрибут max_logname.

Дополнительная информация приведена в описании команды **lsattr** в книге *Справочник по командам, том 3*.

В следующем примере показано применение команды **lsattr** для извлечения атрибута max_logname:

```
$ lsattr -El sys0
SW_dist_intr    false           Включить рассылку SW прерываний                  True
autorestart     true            Автоматически перезагружать систему после сбоя   True
boottype        disk            н/д                                              False
capacity_inc    1.00            Приращение мощности процессора                   False
capped          true            Раздел с ограничениями                           False
conslogin       enable          Вход в систему с консоли                         False
cpuguard        enable          Предохранитель CPU                               True
dedicated       true            Раздел выделенный                                False
ent_capacity    4.00            Предписанная мощность процессора                 False
frequency       93750000        Тактовая частота системной шины                  False
fullcore        false           Включить полный дамп ядра                        True
fwversion       IBM,SPH01316    Версия встроенного ПО и уровни обновления        False
iostat          false           Постоянное ведение хронологии ввода-вывода диска True
keylock         normal          Состояние системного замка во время загрузки     False
max_capacity    4.00            Максимальная возможная мощность процессора       False
max_logname     20              Максимальная длина имени входа во время загрузки True
maxbuf          20              Макс. число страниц в кэше буфера блочного в/в   True
maxmbuf         0               Макс. число Кбайт фактической памяти для MBUFS   True
maxpout         0               Верхняя отметка для ожидающих запросов в/в файла True
maxuproc        128             Максимальное число процессов пользователя        True
min_capacity    1.00            Минимальная возможная мощность процессора        False
minpout         0               Нижняя отметка для ожидающих запросов в/в файла  True
modelname       IBM,7044-270    Имя компьютера                                   False
ncargs          6               Размер списка ARG/ENV в блоках по 4 Кб           True
pre430core      false           Применение дампа ядра стиля до версии 430        True
pre520tune      disable         Режим совместимости с настройкой до версии 520   True
realmem         3145728         Объем доступной физической памяти в Кбайтах      False
rtasversion     1               Версия RTAS открытого встроенного ПО             False
sec_flags       0               Флаги защиты                                     True
```

```
sed_config      select          Режим Отключить обработку стека (SED)          True
systemid        IBM,0110B5F5F   ИД аппаратного обеспечения системы             False
variable_weight 0               Переменный вес мощности процессора             False
$
```

Извлечение ограничения на длину имен пользователей и групп из ядра:

Команды и процедуры позволяют получить параметр **v_max_logname** из ядра.

Применение команды getconf

С помощью команды **getconf** с параметром **LOGIN_NAME_MAX** можно извлечь параметр длины имен пользователей и групп из ядра. Вывод команды **getconf** включает завершающий символ NULL.

В следующем примере показано применение команды **getconf** для извлечения текущего ограничения на длину имен пользователей и групп из ядра:

```
$ getconf LOGIN_NAME_MAX
20
$
```

Применение процедуры sysconf

С помощью процедуры sysconf с параметром **LOGIN_NAME_MAX** можно извлечь параметр длины имен пользователей и групп из ядра.

В следующем примере показано применение процедуры sysconf для извлечения текущего ограничения на длину имен пользователей и групп из ядра:

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

       len = sysconf(_SC_LOGIN_NAME_MAX);

       printf("Максимальная длина имени составляет %d\n", len);
}
```

Применение процедуры sys_parm

С помощью процедуры sys_parm с параметром **SYSP_V_LOGNAME** можно извлечь текущий параметр длины имен пользователей из ядра.

В следующем примере показано применение процедуры sys_parm для извлечения текущего ограничения на длину имен пользователей из ядра:

```
#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("Максимальная длина имени = %d\n", myvar.v.v_max_logname.value);
   else
      printf("Процедура sys_parm() не выполнена rc = %d, errno = %d\n", rc, errno);
}
```

Изменение ограничения на длину имени группы и имени пользователя в базе данных ODM:

Ограничение на длину имени группы и пользователя в ядре можно изменить только во время загрузки системы. Изменить значение в базе данных ODM можно с помощью команды **chdev**. Изменение вступает в силу после перезагрузки системы.

В следующем примере показано применение команды **chdev** для изменения параметра **v_max_logname** в базе данных ODM:

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

Управление учетными записями пользователей:

Атрибуты учетных записей пользователей можно изменять.

С каждой учетной записью пользователя связан набор атрибутов. При создании пользователя с помощью команды **mkuser** этим атрибутам присваиваются значения по умолчанию. Для изменения атрибутов предназначена команда **chuser**. Следующие атрибуты пользователей не применяются для управления паролями:

account_locked
Для явной блокировки учетной записи присвойте этому атрибуту значение True; значение по умолчанию равно False.

admin Значение True запрещает пользователю менять свой пароль. Пароль может изменять только администратор.

admgroups
Список групп, администратором которых является данный пользователь. Администратор может добавлять и удалять пользователей этих групп.

auth1 Способ идентификации пользователя. Обычно равен SYSTEM.

Примечание: Атрибут **auth1** устарел, и использовать его не рекомендуется.

auth2 Способ идентификации, который применяется после идентификации пользователя способом **auth1**. Он не может запрещать доступ пользователя к системе. Обычно равен NONE.

Примечание: Атрибут **auth2** устарел, и использовать его не рекомендуется.

daemon
Эта булевская переменная указывает, может ли пользователь запускать демоны и подсистемы с помощью команды **startsrc**. Также запрещает использование функций cron и at.

login Указывает, разрешено ли пользователю входить в систему. При успешном входе в систему значение атрибута **unsuccessful_login_count** сбрасывается до 0 (из функции **loginsuccess**).

logintimes
Ограничивает время, когда пользователю разрешено входить в систему. Например, пользователю можно разрешить входить в систему только в рабочее время.

registry
Задает реестр пользователя. Может указывать на другие реестры с информацией о пользователе, такие как NIS, LDAP или Kerberos.

rlogin Указывает, может ли заданный пользователь войти в систему с помощью команды **rlogin** или **telnet**. Атрибут контролирует только удаленный вход. Информация об управлении возможностью запуска отдельных удаленных команд приведена в разделе rcmds.

su Указывает, могут ли другие пользователи переключаться на данного с помощью команды **su**.

sugroups
Указывает, каким группам разрешено переключаться на данного пользователя.

ttys Ограничивает набор устройств, с которых пользователь может входить в систему.

expires Управляет гостевыми пользователями; также может применяться для временного отключения пользователя.

loginretries
Задает максимальное количество последовательных неуспешных попыток входа в систему перед блокированием пользователя. Информация о таких попытках записывается в файл `/etc/security/lastlog`.

umask Задает начальное значение **umask** пользователя.

rcmds Указывает, имеет ли указанный пользователь право запуска отдельных команд с использованием команды **rsh** или команды **rexec**. Значение `allow` указывает, что пользователь имеет право запускать удаленные команды с помощью команд **rsh** и **rexec**. Значение `deny` указывает, что право запуска удаленных команд не предоставлено. Значение `hostlogincontrol` указывает, что запуском удаленных команд управляют атрибуты **hostallowedlogin** и **hostsdeniedlogin**. Информация о контроле над удаленным доступом приведена в описании атрибута rlogin.

hostallowedlogin
Задает список хостов, с которых пользователь может войти в систему. Этот атрибут рекомендуется применять в сетевых средах, в которых атрибуты пользователей совместно используются несколькими хостами.

hostsdeniedlogin
Задает список хостов, с которых пользователю запрещено входить в систему. Этот атрибут рекомендуется применять в сетевых средах, в которых атрибуты пользователей совместно используются несколькими хостами.

maxulogs
Задает максимальное число входов каждого пользователя. Если пользователь достиг этого количества, он не сможет войти в систему.

Полный набор атрибутов пользователя определен в файлах `/etc/security/user` ,`/etc/security/limits`, `/etc/security/audit/config` и `/etc/security/lastlog`. Параметры, присваиваемые пользователю при его создании командой **mkuser**, указаны в файле `/usr/lib/security/mkuser.default`. В файле `mkuser.default` следует указывать только значения, которые переопределяют параметры из разделов default в файлах `/etc/security/user` и `/etc/security/limits`. Некоторые из этих атрибутов управляют доступом пользователя к системе и могут применяться для запрета такого доступа при определенных условиях.

Пользователь, заблокированный в результате превышения числа неуспешных попыток входа в систему, не сможет входить в систему до тех пор, пока администратор не сбросит атрибут **unsuccessful_login_count** в файле `/etc/security/lastlog` до значения, меньшего разрешенного числа неудачных попыток входа в систему. Для выполнения этой операции предназначена команда **chsec**:

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

Для изменения значений по умолчанию отредактируйте с помощью команды**chsec** раздел default соответствующего файла защиты, например, `/etc/security/user` or `/etc/security/limits`. Многие значения по умолчанию отражают стандартные ситуации. Для того чтобы изменить атрибуты, присваиваемые пользователю при его создании, измените запись *user* в файле `/usr/lib/security/mkuser.default`.

Информация об атрибутах, связанных с паролями пользователей, приведена в разделе “Пароль” на стр. 64.

Команды, связанные со входом в систему, изменяющиеся с помощью атрибутов пользователя

В следующей таблице приведен список атрибутов, которые контролируют вход в систему и команды.

Атрибут пользователя	Команды
account_locked	**rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet**, **ftp**, **login**
login	Влияет только на вход в систему через консоль. Значение атрибута **login** не влияет на команды удаленного входа в систему, команды удаленных оболочек или копии удаленных команд (**rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet** и **ftp**).
logintimes	**rexec**, **rsh**, **rcp**, **ssh**, **scp**, **rlogin**, **telnet**, **ftp**, **login**
rlogin	Влияет только на команды удаленного входа в систему, некоторые команды удаленной оболочки и некоторые копии удаленных команд (**ssh**, **scp**, **rlogin** и **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**

Примечание: **rsh** только запрещает выполнение удаленных команд. Удаленный вход в систему все еще не запрещен.

Информация, связанная с данной:

Подпрограмма loginsuccess

Команда **rexec**

Команда **rsh**

Команда **startsrc**

Команда **su**

ИД пользователей:

Операционная система распознает пользователей по их идентификаторам (ИД) входа в систему.

Идентификатор позволяет отслеживать все действия, выполняемые пользователем. После входа пользователя в систему, но до запуска начальной программы пользователя система устанавливает в качестве ИД пользователя процесса тот ИД пользователя, который указан в пользовательской базе данных. Кроме того, этот ИД добавляется ко всем последующим процессам, запускаемым в сеансе. Это позволяет отслеживать все операции, выполняемые пользователем. Пользователь может сбросить действующий ИД пользователя, фактический ИД пользователя, действующий ИД группы, фактический ИД группы и дополнительный ИД группы, но он не может изменить ИД, под которым он вошел в систему.

Повышение уровня защиты пользователей с помощью списков управления доступом:

Создайте стратегию управления учетными записями пользователей, позволяющую обеспечить требуемый уровень защиты. Чаще всего для этой цели применяются списки управления доступом.

Информация об ACL и разработке стратегии защиты приведена в разделе “Списки управления доступом” на стр. 123.

Переменная среды PATH:

Переменная среды **PATH** является важной частью системы защиты. Она задает список каталогов, в которых выполняется поиск команд.

Значение **PATH** по умолчанию для всей системы содержится в файле `/etc/profile`, а значения **PATH** для пользователей хранятся в их файлах `$HOME/.profile`. Значение **PATH** в файле `.profile` либо переопределяет системное значение, либо добавляет к нему новые записи.

Несанкционированное изменение **PATH** может дать возможность одним пользователям системы "имитировать" других пользователей (в том числе и пользователей root). Программы *имитации*, известные также как *Троянские кони*, подменяют собой системные команды и собирают предназначенную для этих команд информацию, например, пароли пользователей.

Например, допустим, что пользователь изменил переменную среды **PATH** таким образом, что при поиске команд система сначала просматривает каталог `/tmp`. Затем пользователь поместил в каталог `/tmp` программу с именем **su**, которая запрашивает пароль пользователя root точно так же, как и настоящая команда **su**. Затем программа **/tmp/su** сообщает по электронной почте пароль пользователя root злоумышленнику, вызывает настоящую команду **su** и завершает работу. В этом случае пользователь root, вызвавший команду **su**, раскроет свой пароль, даже не узнав об этом.

Для защиты переменной среды **PATH** от несанкционированного изменения соблюдайте следующие правила:

- При малейшем сомнении указывайте полные имена файлов. В этом случае переменная **PATH** просматриваться не будет.
- Никогда не помещайте текущий каталог (**.**) в переменную **PATH** пользователя root. Не указывайте текущий каталог в файле `/etc/profile`.
- Для пользователя root должна быть задана собственная переменная **PATH** в его личном файле `.profile`. Обычно в файле `/etc/profile` указывается стандартный набор путей для всех пользователей, в то время как пользователю root может требоваться собственный список каталогов поиска.
- Запретите другим пользователям изменять файлы `.profile` без консультации с системным администратором. Таким образом вы обезопасите себя от непреднамеренного создания бреши в системе защиты. Права доступа к файлу `.profile` должны быть равны 740.
- Системным администраторам не следует получать права пользователя root с помощью команды **su** в сеансе другого пользователя, поскольку в этом сеансе используется значение **PATH** из файла `.profile` этого пользователя. Пользователи могут сами настраивать собственные файлы `.profile`. Системные администраторы должны входить в систему пользователя под именем root или, что еще лучше, под собственным именем, а затем вызывать следующую команду:

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

  Это позволит гарантировать, что в сеансе применяется среда root. При работе в качестве root в сеансе других пользователей всегда указывайте полные имена команд и файлов.
- Запретите изменять переменную среды, задающую приглашение на ввод команды (**IFS**), в файле`/etc/profile` file. С помощью переменной среды **IFS** в файле `.profile` можно изменить значение **PATH**.

Использование демона secldapclntd:

Демон **secldapclntd** динамически управляет соединениями с сервером LDAP.

Во время запуска **secldapclntd** устанавливает соединения с серверами, указанными в файле **/etc/security/ldap/ldap.cfg** (по одному соединению на сервер). В дальнейшем, если **secldapclntd** обнаруживает, что соединение ограничивает запросы на обработку LDAP, он автоматически устанавливает еще одно соединение с сервером LDAP. Дополнительные соединения создаются, пока не будет достигнут установленный предел количества соединений. По достижении предела новые соединения больше не создаются.

Демон **secldapclntd** периодически проверяет все соединения с сервером LDAP. Если какое-то (кроме самого первого) соединение не используется, демон закрывает его.

Переменная connectionsperserver в файле **/etc/security/ldap/ldap.cfg** определяет максимальное количество соединений. Однако если значение переменной connectionsperserver больше значения переменной numberofthread, **secldapclntd** приравнивает переменную connectionsperserver к переменной numberofthread. Допустимые значения переменной connectionsperserver: 1—100. Значение по умолчанию — 10 (`connectionsperserver: 10`).

Переменная connectionmissratio в файле **/etc/security/ldap/ldap.cfg** определяет критерии создания новых соединений. Значение переменной connectionmissratio — процент неудавшихся попыток получить соединение LDAP с первого раза. Когда процент неудачных попыток превышает значение переменной connectionmissratio, **secldapclntd** создает дополнительные соединения (их количество ограничено переменной connectionsperserver). Допустимые значения переменной connectionmissratio: 10—90. Значение по умолчанию — 50 (`connectionmissratio: 50`).

Переменная connectiontimeout в файле **/etc/security/ldap/ldap.cfg** определяет разрешенный период простоя соединения, по истечении которого **secldapclntd** закрывает соединение. Допустимые значения: от 5 секунд. Значение по умолчанию — 300 секунд (`connectiontimeout: 300`).

Анонимный FTP с защищенной учетной записью пользователя

Можно настроить анонимный FTP с защищенной учетной записью пользователя.

В этом сценарии продемонстрирована процедура настройки анонимного FTP с защищенной учетной записью пользователя с помощью интерфейса командной строки и сценария.

1. Введите следующую команду, чтобы проверить наличие в системе набора файлов `bos.net.tcp.client`:
   ```
   lslpp -L | grep bos.net.tcp.client
   ```
   Если команда выдаст пустой вывод, значит этот набор файлов не установлен. Инструкции по установке этого набора файлов приведены в книге *Установка и миграция*.
2. Войдите в систему как пользователь root и перейдите в каталог `/usr/samples/tcpip` . Например:
   ```
   cd /usr/samples/tcpip
   ```
3. Для настройки учетной записи запустите следующий сценарий:
   ```
   ./anon.ftp
   ```
4. На вопрос `Вы действительно хотите изменить /home/ftp?`, введите ответ `да`. Вывод команды будет выглядеть примерно следующим образом:
   ```
   Добавлен анонимный пользователь.
   Создан каталог /home/ftp/bin.
   Создан каталог /home/ftp/etc.
   Создан каталог /home/ftp/pub.
   Создан каталог /home/ftp/lib.
   Создана запись /home/ftp/dev/null.
   Создан каталог /home/ftp/usr/lpp/msg/en_US.
   ```
5. Перейдите в каталог `/home/ftp`. Например:
   ```
   cd /home/ftp
   ```
6. Введите следующую команду, чтобы создать каталог `home`:
   ```
   mkdir home
   ```
7. Введите следующую команду, чтобы задать для каталога `/home/ftp/home` права доступа `drwxr-xr-x`:
   ```
   chmod 755 home
   ```
8. Введите следующую команду, чтобы перейти в каталог `/home/ftp/etc`:
   ```
   cd /home/ftp/etc
   ```
9. Создайте каталог `objrepos` с помощью следующей команды:
   ```
   mkdir objrepos
   ```

10. Введите следующую команду, чтобы задать для каталога /home/ftp/etc/objrepos права доступа drwxrwxr-x:

    ```
    chmod 775 objrepos
    ```

11. Измените владельца и группу каталога /home/ftp/etc/objrepos на пользователя root и группу system с помощью следующей команды:

    ```
    chown root:system objrepos
    ```

12. Введите следующую команду, чтобы создать каталог security:

    ```
    mkdir security
    ```

13. Введите следующую команду, чтобы задать для каталога /home/ftp/etc/security права доступа drwxr-x:

    ```
    chmod 750 security
    ```

14. Измените владельца и группу каталога /home/ftp/etc/security на пользователя root и группу security с помощью следующей команды:

    ```
    chown root:security security
    ```

15. Введите следующую команду, чтобы перейти в каталог /home/ftp/etc/security:

    ```
    cd security
    ```

16. Добавьте пользователя с помощью следующей команды быстрого доступа SMIT:

    ```
    smit mkuser
    ```

    В этом сценарии добавляется пользователь с именем test.

17. Укажите в полях SMIT следующие значения:

    ```
    Имя пользователя [test]
    Пользователь администратор?                                 истина
    Основная группа                                  [staff]
    Набор групп                                      [staff]
    Может ли другой пользователь сделать SU к пользователю?                        истина
    Домашний каталог                                 [/home/test]
    ```

    После внесения необходимых изменений нажмите клавишу Enter, чтобы добавить этого пользователя. По завершении процесса SMIT выйдите из интерфейса SMIT.

18. Создайте пароль для этого пользователя с помощью следующей команды:

    ```
    passwd test
    ```

    Когда появится приглашение, введите выбранный пароль. Новый пароль необходимо ввести еще раз для подтверждения.

19. Введите следующую команду, чтобы перейти в каталог /home/ftp/etc:

    ```
    cd /home/ftp/etc
    ```

20. Скопируйте файл /etc/passwd в файл /home/ftp/etc/passwd с помощью следующей команды:

    ```
    cp /etc/passwd /home/ftp/etc/passwd
    ```

21. Откройте файл /home/ftp/etc/passwd в любом текстовом редакторе. Например:

    ```
    vi passwd
    ```

22. Удалите из скопированного содержимого файла все строки, кроме пользователей root, ftp и test. После внесения изменений содержимое файла должно выглядеть следующим образом:

    ```
    root:!:0:0::/:/bin/ksh
    ftp:*:226:1::/home/ftp:/usr/bin/ksh
    test:!:228:1::/home/test:/usr/bin/ksh
    ```

23. Сохраните изменения и закройте редактор.

24. Введите следующую команду, чтобы задать для файла /home/ftp/etc/passwd права доступа -rw-r--r--:

    ```
    chmod 644 passwd
    ```

25. Измените владельца и группу файла /home/ftp/etc/passwd на пользователя root и группу security с помощью следующей команды:

    ```
    chown root:security passwd
    ```

26. Скопируйте содержимое файла /etc/security/passwd в файл /home/ftp/etc/security/passwd с помощью следующей команды:

    ```
    cp /etc/security/passwd /home/ftp/etc/security/passwd
    ```

27. Откройте файл /home/ftp/etc/security/passwd в любом текстовом редакторе. Например:

    ```
    vi ./security/passwd
    ```

28. Удалите из файла все записи кроме записи пользователя test.
29. Удалите строку flags = ADMCHG из записи пользователя test. После внесения изменений содержимое файла должно выглядеть следующим образом:

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

30. Сохраните изменения и закройте редактор.
31. Введите следующую команду, чтобы задать для файла /home/ftp/etc/security/passwd права доступа -rw-------:

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

32. Измените владельца и группу файла /home/ftp/etc/security/passwd на пользователя root и группу security с помощью следующей команды:

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

33. Создайте и откройте файл /home/ftp/etc/group в любом текстовом редакторе. Например:

    ```
    vi group
    ```

34. Добавьте в этот файл следующие строки:

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

35. Сохраните изменения и закройте редактор.
36. Введите следующую команду, чтобы задать для файла /home/ftp/etc/group права доступа -rw-r--r--:

    ```
    chmod 644 group
    ```

37. Измените владельца и группу файла /home/ftp/etc/group на пользователя root и группу security с помощью следующей команды:

    ```
    chown root:security group
    ```

38. Создайте и откройте файл /home/ftp/etc/security/group в любом текстовом редакторе. Например:

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

39. Добавьте в этот файл следующие строки:

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

40. Сохраните изменения и закройте редактор. Это можно сделать следующим образом:
    a. Скопируйте файл /etc/security/user в каталог /home/ftp/etc/security, введя команду:

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

    b. Удалите в редакторе все разделы из скопированного содержания, кроме раздела для пользователя test, введя команду:

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

    c. Сохраните и закройте редактор.
41. Введите следующую команду, чтобы задать для файла /home/ftp/etc/security/group права доступа -rw-r-----:

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

42. Измените владельца и группу файла /home/ftp/etc/security/group на пользователя root и группу security с помощью следующей команды:

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

43. С помощью следующих команд добавьте необходимые данные в каталог /home/ftp/etc/objrepos:

    ```
    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. Введите следующую команду, чтобы перейти в каталог /home/ftp/home:

    ```
    cd ../home
    ```

45. Создайте новый домашний каталог для этого пользователя с помощью команды:

    ```
    mkdir test
    ```

    Этот каталог будет домашним каталогом нового пользователя ftp.

46. Измените владельца и группу каталога /home/ftp/home/test на пользователя test и группу staff с помощью следующей команды:

    ```
    chown test:staff test
    ```

47. Задайте для файла /home/ftp/home/test права доступа -rwx------ с помощью следующей команды:

    ```
    chmod 700 test
    ```

48. Отключите удаленный вход и вход с консоли для пробного пользователя, введя:

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

Вы настроили вход в систему с помощью ftp. Его можно проверить, выполнив следующую процедуру:

1. Установите соединение ftp с системой, в которой был создан пользователь test. Например:

   ```
   ftp MyHost
   ```

2. Войдите в систему как пользователь anonymous. Когда появится приглашение на ввод пароля, нажмите клавишу Enter.
3. Переключитесь на созданного пользователя test с помощью следующей команды:

   ```
   user test
   ```

   В приглашении на ввода пароля введите пароль, заданный на шаге 18 на стр. 58

4. Проверьте с помощью команды **pwd** наличие домашнего каталога этого пользователя. Например:

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

   В выводе каталог /home/test будет указан как подкаталог ftp. Фактическое полное имя каталога в системе - /home/ftp/home/test.

Примечания:

- Можно переключать только пользователей ftp sub. Например, test - это это пользователь ftp sub.
- При создании пользователей ftp anonymous с помощью сценария anon.users.ftp можно назначить пользователю любое имя, заменив *username* в сценарии.
- Так как сервер выполняет команду **chroot** в домашнем каталоге учетной записи пользователя, для пользователей anonymous все файлы, относящиеся к конфигурации, такие как *fileftpaccess.ctl*, должны находиться в домашнем каталоге, таком как ~/etc/, соответствующего анонимного пользователя. Пути ограничений 'writeonly', 'readonly' и 'readwrite' в файле /etc/ftpaccess.ctl должны быть заданы относительно пути chrooted.

Дополнительная информация:

- Раздел "Защита TCP/IP" в книге *Защита*
- Раздел "**Команда ftp**" в книге *Справочник по командам*

Специальные системные учетные записи

В AIX существует набор специальных учетных записей по умолчанию, позволяющий не делать пользователей root и system владельцами всех системных файлов и файловых систем.

Внимание: При удалении специальных учетных записей следует соблюдать особую осторожность. Для отключения учетной записи укажите звездочку (*) в начале соответствующей строки файла `/etc/security/passwd`. Учетную запись root отключать нельзя. Если вы удалите специальные системные учетные записи или отключите учетную запись root, то операционная система работать не сможет.

В операционной системе предопределены следующие учетные записи:

adm Пользователю adm принадлежат следующие основные системные функции:
- Средства диагностики, хранящиеся в каталоге `/usr/sbin/perf/diag_tool`.
- Средства учета, хранящиеся в следующих каталогах:
  - `/usr/sbin/acct`
  - `/usr/lib/acct`
  - `/var/adm`
  - `/var/adm/acct/fiscal`
  - `/var/adm/acct/nite`
  - `/var/adm/acct/sum`

bin Учетной записи bin обычно принадлежат исполняемые файлы большинства команд. Ее основное предназначение - помочь распределить права владения важными системными каталогами и файлами, чтобы не вся система принадлежала пользователям root и sys.

daemon
Учетная запись daemon применяется только для запуска процессов системного сервера и в качестве владельца связанных файлов. Она позволяет гарантировать, что такие процессы будут работать с правильными правами доступа.

nobody
Учетная запись nobody применяется сетевой файловой системой (NFS) для удаленной печати. Она позволяет временно предоставить доступ root пользователям root. Например, перед тем как включить Защищенный RPC или Защищенный NFS, найдите в ключе `/etc/public` на сервере NIS пользователя, которому не был присвоен общий или частный ключ. Пользователь root может создать запись в базе данных для каждого пользователя без подписи:

```
newkey -u имя_пользователя
```

Если же вы создадите в базе данных запись для пользователя nobody, то любой пользователь сможет создавать для себя записи с помощью программы **chkey**, не входя в систему как root.

root Учетная запись пользователя root с UID 0 применяется для обслуживания системы и устранения неполадок.

sys Пользователю sys принадлежит точка монтирования кэша распределенной файловой системы (DFS) по умолчанию, которая должна существовать до установки или настройки DFS на клиенте. Кроме того, в каталоге `/usr/sys` хранятся установочные образы.

system Группа system - это системная группа, в состав которой входят администраторы. Пользователи группы system обладают правами доступа на выполнение различных задач обслуживания системы. Для этого не требуются права доступа root.

Удаление стандартных учетных записей пользователей, в которых нет необходимости:

При установке операционной системы создается несколько пользователей и групп по умолчанию. В зависимости от набора приложений в вашей системе и расположения системы в сети, некоторые из этих пользователей могут стать потенциально уязвимым местом в системе защиты.

В следующей таблице перечислены основные ИД пользователей по умолчанию, которые может потребоваться удалить:

Таблица 3. Основные ИД пользователей по умолчанию, которые может потребоваться удалить.

ИД пользователя	Описание
uucp, nuucp	Владелец скрытых файлов, используемых протоколом uucp. Учетная запись uucp применяется в протоколе UNIX-to-UNIX Copy Program, который представляет собой набор команд, программ и файлов, имеющихся в большинстве систем AIX, для взаимодействия с другими системами AIX по выделенной или телефонной линии.
lpd	Владелец файлов в подсистеме печати.
guest	Предоставляет доступ пользователям, не имеющим учетных записей.

В следующей таблице перечислены некоторые ИД групп, которые может потребоваться удалить:

Таблица 4. Некоторые ИД групп, которые может потребоваться удалить.

ИД группы	Описание
uucp	Группа пользователей uucp и nuucp
printq	Группа пользователя lpd.

Выясните, какие из перечисленных ИД не нужны в системе. Лишними могут оказаться и другие ИД пользователей и групп. Удалите все ненужные ИД до начала полноценной работы с системой.

Примечание: Вместо удаления группы `printq` вследствие зависимости от наборов файлов принтера, выключите ИД пользователя `lp`, команду и программу **piobe** command, and the `qdaemon` в записи `/etc/inittab`, чтобы свести к минимуму угрозы безопасности. В этом случае пользователь не сможет выполнять команды **print**.

Учетные записи, созданные компонентами защиты:

При установке или настройке таких компонентов защиты, как LDAP и OpenSSH создаются учетные записи пользователей и групп.

Созданные учетные записи пользователей и групп включают:

- **Защита протоколов Internet (IP):** Защита IP добавляет пользователя *ipsec* и группу *ipsec* во время установки. Эти ИД используются службой управления ключами. Заметьте, что ИД группы в `/usr/lpp/group.id.keymgt` нельзя настроить перед установкой.
- **Kerberos и инфраструктура общих ключей (PKI):** Данные компоненты не создают новых учетных записей пользователей и групп.
- **LDAP:** При установке клиента или сервера LDAP, создается учетная запись пользователя *ldap*. ИД пользователя *ldap* не является постоянным. Во время установки сервера LDAP автоматически устанавливается база данных DB2. При установке DB2 создается учетная запись группы *dbsysadm*. По умолчанию ИД группы *dbsysadm* 400. Во время настройки сервера LDAP команда **mksecldap** создает учетную запись пользователя *ldapdb2*.
- **OpenSSH:** При установке OpenSSH, в систему добавляется пользователь *sshd* и группа *sshd*. Соответствующие ИД пользователя и группы изменять нельзя. Для функции разделения прав доступа в SSH необходимы ИД.

Группы без доменов

Компонент групп без доменов предоставляет возможность помещать пользователей, определенных в одном домене, в группы, определенные в другом домене. Этот компонент поддерживает только домены LDAP и локальные домены.

Пользователи и группы на сервере LDAP создаются с помощью загружаемого модуля идентификации LDAP (модуля LDAP). Также можно создавать пользователей и группы в локальной системе с помощью загружаемого модуля локальной идентификации (локального модуля). Когда компонент **domainlessgroups** выключен, невозможно помещать пользователей и группы пользователей, которые созданы в LDAP или в локальной системе, в группы, размещенные в другом домене. Например, пользователя, который создается в домене LDAP, невозможно поместить в группу, связанную с локальным доменом.

Для того чтобы устранить это ограничение и помещать пользователей как в группы LDAP, так и в локальные группы, включите системное свойство **domainlessgroups**. Свойство **domainlessgroups** определено в файле `/etc/secvars.cfg`. Оно поддерживается только для модулей LDAP локальных модулей. Ниже приведены допустимые значения этого свойства:

false (по умолчанию)
: Атрибут группы из модулей LDAP и локальных модулей не включается.

истина
: Атрибут группы из модулей LDAP и локальных модулей включается. Например, пользователей LDAP можно поместить в локальные группы.

Для просмотра значения свойства **domainlessgroups** запустите следующую команду:

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

Для того чтобы присвоить свойству **domainlessgroups** значение true, запустите следующую команду:

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

В приведенной ниже таблице показаны различия между командами для пользователей и групп в зависимости от значения свойства **domainlessgroups**.

Таблица 5. Результаты выбранных команд, на которые влияет свойство **domainlessgroups**

Команда	Результаты при значении свойства domainlessgroups true
`chgroup -R ldap|files`	Обновляет группу в указанном домене. Пользователя можно добавить как в группу LDAP, так и в локальную группу.
`chuser -R ldap|files`	Изменяет параметры пользователя в указанном домене. Если указаны группы, определенные в другом домене, то информация о пользователях в этих группах тоже обновляется.
`login` *username* или `su`	Извлекает атрибуты пользователя из реестра пользователей за исключением атрибута ИД группы. Включаются атрибуты пользователя для ИД группы как из домена LDAP, так и из локального домена.
`lsgroup -R ldap|files`	Выводит все атрибуты группы для указанного домена. Если указанная группа не найдена в указанном домене, то происходит сбой команды.
`lsuser -R ldap|files`	Выводит атрибуты пользователя после объединения информации из всех групп домена, в котором определен пользователь, и из другого домена. Если основная группа пользователя не определена в домене, в котором определен пользователь, оно извлекается из другого домена.
`mkgroup -R ldap|files`	Создает группу в указанном домене. После создания группы пользователь (LDAP или локальный) помещается в базу данных групп этого домена. Пользователя можно добавить в группу LDAP или в локальную группу.

Таблица 5. Результаты выбранных команд, на которые влияет свойство **domainlessgroups** *(продолжение)*

Команда	Результаты при значении свойства domainlessgroups true
`mkuser -R ldap|files`	Создает пользователя в указанном домене. Если указаны группы, определенные в другом домене, то информация о пользователях в этих группах тоже обновляется.
`rmgroup -R ldap|files`	Удаляет указанную группу из указанного домена. Если группа назначается основной группой какого-либо пользователя, определенного в каком-либо домене, то происходит сбой команды.
`rmuser -R ldap|files`	Удаляет указанного пользователя из указанного домена. Также удаляет пользователя из всех групп, определенных в другом домене, если он является их участником.

Понятия, связанные с данным:

“Загружаемый модуль идентификации LDAP” на стр. 156
Взаимодействие LDAP с подсистемой защиты осуществляется через загрузочный модуль идентификации. Этот модуль схож с аналогичными загрузочными модулями NIS, DCE и KRB5 5. Определения загрузочных модулей хранятся в файле `/usr/lib/security/methods.cfg`.

Информация, связанная с данной:

Команда chgroup

Команда chuser

login, команда

lsgroup, команда

Команда lsuser

Команда mkgroup

Команда mkuser

Команда rmgroup

Команда rmuser

su, команда

Пароль

Угадывание паролей - один из наиболее распространенных способов проникновения в систему. В этой связи особую роль приобретают разработка и управление стратегией паролей.

В AIX предусмотрены средства, позволяющие усиливать стратегию паролей, в частности, устанавливать следующие значения:

- Минимальную и максимальную продолжительность действия пароля в неделях
- Минимальную длину пароля
- Минимальное число букв в пароле

Выбор надежных паролей:

Надежные пароли обеспечивают первую линию защиты от несанкционированного доступа в систему.

Пароли считаются надежными, если они:

- Содержат как строчные, так и прописные буквы
- Содержат буквы, цифры и знаки препинания. Кроме того, пароли могут содержать специальные символы: `~!@#$%^&*()-_=+[]{}|\;:'",.<>?/<пробел>`
- Нигде не записаны
- Длина составляет от 7 до PW_PASSLEN символов, если применяется файл `/etc/security/passwd` (ограничение сверху на длину пароля не распространяется на реализации функции идентификации на основе реестров, например LDAP)

- Не являются обычными словами, которые можно найти в любом словаре
- Никак не связаны с расположением клавиш на клавиатуре (пример неудачного пароля - *qwerty*)
- Не являются обычными словами или известными шаблонами, написанными в обратном порядке
- Не содержат личную информацию о пользователе, его семье или друзьях
- Составлены по другому принципу, нежели предыдущие пароли
- Могут быть введены достаточно быстро, чтобы находящиеся поблизости люди не могли определить пароль

Помимо этих ограничений, можно установить еще более строгие правила, запретив использование в паролях стандартных слов UNIX, которые можно угадать. Для этого случая используется словарь dictionlist, для применения которого необходимо предварительно установить наборы файлов `bos.data` и `bos.txt`.

Для использования существующего словаря dictionlist измените в файле `/etc/security/users` следующую строку:

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

Файл `/usr/share/dict/words` позволяет с помощью словаря dictionlist запретить использование стандартных слов UNIX в качестве паролей.

Файл /etc/passwd:

Традиционно в файле `/etc/passwd` сохраняются данные обо всех зарегистрированных пользователях, имеющих доступ к системе.

Файл `/etc/passwd` содержит следующие записи, разделенные двоеточиями:

- Имя пользователя
- Зашифрованный пароль
- Цифровой идентификатор пользователя (UID)
- Цифровой идентификатор группы пользователя (GID)
- Полное имя пользователя (GECOS)
- Домашний каталог пользователя
- Оболочка входа в систему

Ниже приведен пример файла `/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
```

В отличие от систем UNIX, система AIX хранит зашифрованные пароли не в файле `/etc/password`, а в файле `/etc/security/password`[1], который доступен для чтения только пользователю root. Значение в поле пароля в файле `/etc/passwd` в AIX просто указывает, задан ли пароль и заблокирована ли учетная запись.

1. /etc/security/password

Файл /etc/passwd принадлежит пользователю root и должен быть доступен для чтения всем пользователям, но для записи - только пользователю root, на что и указывают права доступа -rw-r--r--. Если для ИД пользователя установлен пароль, то в поле пароля будет находиться ! (восклицательный знак). Если у пользователя нет пароля, то в поле пароля будет указана звездочка (*). Зашифрованные пароли хранятся в файле /etc/security/passwd. Приведенный ниже пример содержит четыре последних записи файла /etc/security/passwd, соответствующего приведенному выше файлу /etc/passwd.

```
guest:
        password = *

nobody:
        password = *

lpd:
        password = *

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

У пользователя jdoe нет записи в файле /etc/security/passwd, так как для него не задан пароль в файле /etc/passwd.

Соответствие файла /etc/passwd можно проверить с помощью команды**pwdck**. Команда **pwdck** проверяет указанную в файлах пользовательской базы данных информацию о паролях, просматривая определения всех или только указанных пользователей.

Файл /etc/passwd и сетевые среды:

В традиционной сетевой среде учетная запись пользователя создавалась в каждой системе, к которой ему предоставлялся доступ.

Это означало, что в каждой из таких систем файл /etc/passwd содержал запись для этого пользователя. Однако в распределенной среде сложно обеспечить идентичность файлов /etc/passwd во всех системах. Для решения этой проблемы были разработаны различные средства, обеспечивающие сетевой доступ к содержимому файла /etc/passwd, включая службу информации о сети (NIS).

Скрытие имен пользователей и паролей:

Для повышения надежности защиты необходимо скрыть ИД пользователей и пароли в системе.

Файлы .netrc содержат ИД пользователей и пароли. Данные в этих файлах представлены в виде обычного незашифрованного текста. Для того чтобы найти эти файлы, введите следующую команду:

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

Обнаружив эти файлы, удалите их. Более надежный способ защитить пароли - настроить Kerberos. Дополнительная информация о Kerberos приведена в разделе “Kerberos” на стр. 291.

Настройка рекомендуемых опций паролей:

Правильной работы с паролями можно достичь только за счет обучения пользователей. Для обеспечения дополнительной защиты в операционной системе можно настроить ограничения на пароли. С их помощью администратор может установить правила для выбираемых пользователями паролей и задать частоту их смены.

Опции паролей и расширенные атрибуты пользователей находятся в текстовом файле /etc/security/user, который содержит разделы атрибутов пользователей. Эти ограничения применяются при определении нового пароля пользователя. Ограничения определяются для каждого пользователя отдельно. Ограничения,

заданные в разделе атрибутов по умолчанию в файле /etc/security/user, распространяются на всех пользователей. Для обеспечения защиты паролей необходимо задать схожие ограничения для всех паролей.

Администраторы также могут изменять ограничения. С помощью атрибута **pwdchecks** файла/etc/security/user администратор может добавить новые функции (*методы*) в код проверки ограничений на пароли. Таким образом, локальные стратегии можно установить и применять в операционной системе. Дополнительная информация приведена в разделе “Расширение ограничений на пароли” на стр. 71.

Ограничения на пароли следует устанавливать разумно. Попытки установить слишком строгие ограничения могут существенно снизить эффективность защиты с помощью паролей: например, ограничение пространства паролей упростит их угадывание, а требование выбирать труднозапоминаемые пароли приведет к тому, что пользователи будут их записывать. В конечном счете, надежность защиты с помощью пароля зависит от пользователя. Простые ограничения на пароли, разумные рекомендации по выбору новых паролей и периодическая проверка уникальности паролей - вот основные составляющие оптимальной стратегии.

Следующая таблица содержит рекомендуемые значения для некоторых атрибутов защиты, связанных с паролями, в файле /etc/security/user.

Таблица 6. Рекомендуемые значения атрибутов защиты для паролей пользователей.

Атрибут	Описание	Рекомендуемое значение	Значение по умолчанию	Максимальное значение
dictionlist	Проверяет пароли на отсутствие стандартных слов UNIX.	**/usr/share/dict/words**	неприменимо	неприменимо
histexpire	Время в неделях, по истечении которого пароль может применяться повторно.	26	0	260*
histsize	Разрешенное количество повторений пароля.	20	0	50
maxage	Максимальная продолжитель- ность действия пароля в неделях.	8	0	52
maxexpired	Максимальное время в неделях сверх *maxage*, в течение которого пароль может быть изменен пользователем. (Не распростра- няется на пользователя root.)	2	-1	52
maxrepeats	Максимальное число повторяющихся символов в пароле.	2	8	8

Таблица 6. Рекомендуемые значения атрибутов защиты для паролей пользователей. (продолжение)

Атрибут	Описание	Рекомендуемое значение	Значение по умолчанию	Максимальное значение
minage	Минимальная продолжитель- ность действия пароля. Присваивать ненулевое значение этому атрибуту рекомендуется только в том случае, если всегда можно обратиться к администратору для изменения недавно измененного пароля, который был случайно раскрыт.	0	0	52
minalpha	Минимальное число букв в пароле.	2	0	PW_PASSLEN**
mindiff	Минимальное число уникальных символов в пароле.	4	0	PW_PASSLEN**
minlen	Минимальная длина пароля.	6 (8 - для пользователя root)	0	PW_PASSLEN**
minother	Минимальное число символов, отличных от букв, в пароле.	2	0	PW_PASSLEN**
pwdwarntime	Время в днях, по истечении которого система выдает предупреждение о необходимости изменения пароля.	5	неприменимо	неприменимо
pwdchecks	Эта запись позволяет добавить в команду **passwd** пользовательский код проверки качества пароля.	Дополнительная информация приведена в разделе “Расширение ограничений на пароли” на стр. 71.	неприменимо	неприменимо

* Сохраняется не более 50 паролей.

** PW_PASSLEN определяется в `userpw.h`

Если в системе установлены функции обработки текстов, то в качестве файла словаря **dictionlist** администратор может взять файл `/usr/share/dict/words`. В этом случае администратор может присвоить атрибуту **minother** значение `0`. Так как большинство слов в словаре не содержат символов, попадающих в категорию атрибута **minother**, то присвоение атрибуту **minother** значения 1 или больше делает подавляющее большинство слов в этом словаре ненужными.

Минимальная длина пароля в системе определяется большим из двух значений: значения атрибута **minlen** и суммы значений атрибутов **minalpha** и **minother**.

Максимальная длина пароля составляет число символов, указанных в атрибуте **PW_PASSLEN**. Число символов, используемых при создании сохраняемого значения пароля зависит от алгоритма создания паролей, используемого в системе. Алгоритмы формирования паролей определены в файле `/etc/security/pwdalg.cfg`, а алгоритм для использования по умолчанию можно настроить с помощью атрибута **pwd_algorithm** в файле `/etc/security/login.cfg`. Сумма значений атрибутов **minalpha** и **minother** ни в коем случае не должна превышать значение атрибута **PW_PASSLEN**. Если сумма значений атрибутов

minalpha и **minother** превышают значение атрибута **PW_PASSLEN**, то значение атрибута **minother** уменьшается до значения **PW_PASSLEN** минус значение **minalpha**.

Если заданы значения атрибутов **histexpire** и **histsize**, то система сохраняет число паролей, удовлетворяющее обоим условиям, но не свыше 50 паролей для каждого из пользователей. Пустые пароли не сохраняются.

С помощью текстового редактора в файл `/etc/security/user` можно внести произвольные значения по умолчанию для управления паролями пользователей. Кроме того, изменить значения атрибутов можно с помощью команды **chuser**.

Помимо данной, для этого файла можно запускать команды **mkuser**, **lsuser** и **rmuser**. Команда **mkuser** создает запись для каждого нового пользователя в файле `/etc/security/user` и присваивает его атрибутам значения из файла `/usr/lib/security/mkuser.default`. Команда **lsuser** предназначена для просмотра атрибутов и соответствующих значений. Команда **rmuser** позволяет удалить пользователя.

Поддержка паролей длиной более 8 символов и загружаемых алгоритмов формирования паролей:

В свете последних достижений в области аппаратного обеспечения традиционное шифрование паролей UNIX стало уязвимым для атак методом подбора пароля. Слабый с криптографической точки зрения алгоритм может привести к раскрытию даже сложных паролей. AIX поддерживает загружаемый алгоритм формирования паролей (LPA), который предоставляет защищенные механизмы хэширования паролей.

Традиционная функция crypt шифрования пароля:

Стандартный механизм идентификации AIX использует функцию необратимого шифрования **crypt** для проверки подлинности пользователей. Функция **crypt** применяет модифицированный алгоритм DES. Она выполняет необратимое шифрование массива фиксированной длины для пароля и дополнительной строки.

Функция **crypt** использует только первые восемь символов из пароля, остальные символы отбрасываются. Если пароль содержит менее восьми символов, то он дополняется нулями справа. На основе 7 битов из каждого символа вычисляется 56-разрядный ключ DES.

В алгоритме DES используются 12 битов дополнительной строки. Дополнительная строка - это два символа из набора "**A-Z**", "**a-z**","**0-9**","**.**" (точка) и "/". Дополнительная строка позволяет усложнить алгоритм хэша, и один и тот же пароль может порождать 4096 зашифрованных строк. Кроме того, в алгоритме DES меняются местами биты i и i+24 в выводе DES E-Box, когда бит i задан в дополнительной строке, что устраняет возможность использования аппаратных средств шифрования DES для подбора пароля.

Ключом DES 25 раз шифруется 64-разрядный нулевой битовый массив. Выводом является строка, составленная из 12 бит дополнительной строки и зашифрованного 64-разрядного битового массива. Получившиеся 76 бит преобразуются в 13 символов ASCII в формате base64.

Алгоритмы хэширования паролей:

Такие алгоритмы хэширования паролей, как MD5, тяжелее поддаются взлому, чем функция **crypt**. Это обеспечивает надежный механизм защиты от атак путем подбора паролей. Поскольку для создания хэша используется весь пароль, при использовании алгоритмов хэширования для шифрования паролей отсутствуют ограничения на длину пароля.

Загружаемый алгоритм шифрования паролей:

AIX 6.1 и более поздних версий реализует механизм загружаемого алгоритма формирования паролей (LPA), который может с легкостью развертывать новые алгоритмы шифрования паролей.

Каждый поддерживаемый алгоритм шифрования пароля реализуется как загружаемый модуль LPA, который загружается во время выполнения, когда возникает необходимость в алгоритме. Поддерживаемые LPA и их атрибуты определены в файле конфигурации системы /etc/security/pwdalg.cfg.

Администратор может установить механизм шифрования паролей для всей системы, который использует для шифрования паролей особый LPA. После изменения общесистемного механизма шифрования паролей, пароли, зашифрованные старыми методами (например, функцией **crypt**), все равно будут поддерживаться.

Поддержка паролей длиной более восьми символов:

Все LPA, реализованные для AIX 6.1 и выше, поддерживают пароли длиннее 8 символов. Для различных LPA существуют различные ограничения на длину пароля. Максимальная поддерживаемая длина пароля — 255 символов.

Файл конфигурации LPA:

Файлом конфигурации LPA является /etc/security/pwdalg.cfg. Это файл настройки, в котором определены атрибуты поддерживаемых LPA.

В файле настройки определены следующие атрибуты:

- Путь к модулю LPA
- Необязательные флаги, которые передаются модулю LPA во время выполнения

К атрибутам LPA, определенным в файле настройки, можно обращаться через интерфейсы **getconfattr** и **setconfattr**.

В следующем файле настройки /etc/security/pwdalg.cfg определен LPA с именем **ssha256**:

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

Алгоритм паролей системы:

Системный администратор может задать общесистемный алгоритм паролей, выбрав LPA в качестве алгоритма хеширования паролей. В каждый момент времени может быть активен только один алгоритм паролей системы. Алгоритм паролей системы определяется системным атрибутом **pwd_algorithm** в разделе **usw** файла /etc/security/login.cfg.

Допустимые значения атрибута **pwd_algorithm** в файле /etc/security/login.cfg - это имена раздела LPA, определенные в файле /etc/security/pwdalg.cfg. Другое допустимое значение для атрибута **pwd_algorithm** - это **crypt**, соответствующее традиционному шифрованию **crypt**. Если атрибут **pwd_algorithm** опущен в файле конфигурации, то в качестве значения по умолчанию применяется **crypt**.

В следующем примере файла /etc/security/login.cfg в качестве общесистемного алгоритма шифрования паролей применяется LPA **ssha256**:

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

Алгоритм паролей системы действует только для вновь создаваемых и изменяемых паролей. После переноса все последующие новые или измененные пароли создаются согласно этому алгоритму. Пароли,

существовавшие до выбора данного алгоритма, - созданные стандартной функцией **crypt** или другими поддерживаемыми модулями LPA, - по-прежнему будут работать в системе. Таким образом, в системе могут сосуществовать пароли, созданные различными LPA.

Настройка алгоритма паролей системы:

Системный администратор может настроить алгоритм паролей системы с помощью команды **chsec** или вручную изменить атрибут **pwd_algorithm** в файле `/etc/security/login.cfg` с помощью редактора, такого как **vi**.

Настраивать алгоритм паролей системы рекомендуется командой **chsec**, поскольку команда **chsec** автоматически проверяет определение указанного LPA.

Работа с командой chsec

Для задания LPA **smd5** в качестве общесистемного модуля шифрования паролей выполните следующую команду:

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

Когда вы изменяете атрибут **pwd_algorithm** с помощью команды **chsec**, команда **chsec** проверяет указанный LPA в файле `/etc/security/pwdalg.cfg`. Если проверка окажется неудачной, команда **chsec** выполнена не будет.

Работа с редактором

Если вы вручную изменяете значение атрибута **pwd_algorithm** в файле `/etc/security/login.cfg` с помощью редактора, то убедитесь, что указанное значение есть имя раздела, определенного в файле `/etc/security/pwdalg.cfg`.

Расширение ограничений на пароли:

К правилам, согласно которым программа паролей принимает или отклоняет пароли (ограничения на составление паролей), системные администраторы могут добавить локальные ограничения.

Расширение ограничений выполняется путем добавления функций, называемых методами, которые вызываются при изменении паролей. Вызываемые методы задаются атрибутом **pwdchecks** в файле `/etc/security/user`.

Описание интерфейса функции **pwdrestrict_method**, которому должны соответствовать указанные методы ограничений на пароли, приведено в книге *AIX версии 6.1: Технический справочник*. Для того чтобы правильно расширить ограничения на составление паролей, системный администратор должен использовать этот интерфейс при написании метода ограничения паролей. При расширении ограничений на составление паролей следует соблюдать осторожность. Эти дополнения непосредственно влияют на работу команд **login**, **passwd**, **su** и других программ. Умышленные или неумышленные искажения кода легко могут привести к нарушению защиты системы.

Идентификация пользователей

Идентификация позволяет выяснить, является ли пользователь тем, за кого он себя выдает.

Идентификация выполняется при входе в систему. Вы указываете свой ИД пользователя и пароль, если он определен для учетной записи (в защищенной системе пользователям без пароля должно быть запрещено выполнение важных операций). Если пароль введен правильно, то вам назначается учетная запись: вы получаете права доступа этой учетной записи. Пароли пользователей хранятся в файле `/etc/passwd` и `/etc/security/passwd`.

По умолчанию пользователи определяются в файлах реестра. Такой подход предусматривает хранение информации об учетных записях пользователей и групп в обычных текстовых файлах ASCII. Встраиваемые модули позволяют добавлять сведения о пользователях и в другие реестры. Например, если для администрирования пользователей применяется встраиваемый модуль LDAP, определения пользователей хранятся в каталоге LDAP. В этом случае записи пользователей в файл `/etc/security/user` не добавляются (исключение составляют только атрибуты пользователей **SYSTEM** и **registry**). Если для администрирования пользователей применяется составной модуль (например, модули, обеспечивающие идентификацию и взаимодействие с базой данных), то способ управления информацией об учетных записях пользователей AIX, а также процесс управления идентификацией и паролями описываются различными частями этого модуля. Кроме того, вы можете указать атрибуты администрирования пользователей, связанные с идентификацией, путем реализации соответствующих интерфейсов модулей (newuser, getentry, putentry и т.д.)

Метод идентификации управляется атрибутами **SYSTEM** и реестра, которые определяются в файле `/etc/security/user`. Системный администратор может задать атрибут `authcontroldomain` в файле `/etc/security/login.cfg` на применение атрибутов **SYSTEM** и реестра, извлекаемых из `authcontroldomain`. Например, `authcontroldomain=LDAP` может указывать системе поиск атрибутов **SYSTEM** и реестра пользователя в LDAP для определения метода идентификации, применяемого для данного пользователя. Существует исключение для локально определенных пользователей, для которых параметр `authcontroldomain` игнорируется, и атрибуты **SYSTEM** и реестра всегда извлекаются из файла `/etc/security/user`.

Приемлемым ключом для атрибута `authcontroldomain` являются файлы или имя раздела из файла `/usr/lib/security/methods.cfg`.

Значение атрибута **SYSTEM** определяется в соответствии с грамматикой. С помощью этой грамматики системные администраторы могут комбинировать один или более методов идентификации конкретного пользователя в системе. Наиболее часто применяются следующие маркеры методов: `compat`, `DCE`, `files` и `NONE`.

По умолчанию применяется значение `compat`. Атрибут `SYSTEM=compat` указывает системе, что для определения имен (и последующей идентификации) применяется локальная база данных, а затем - база данных NIS. Значение `files` указывает, что будут применяться только локальные файлы, `SYSTEM=DCE` указывает на применение потока идентификации `DCE`.

Значение `NONE` отключает все способы идентификации. Для полного отключения идентификации укажите значение `NONE` в строках `SYSTEM` и `auth1` раздела пользователя.

При необходимости вы можете указать несколько способов идентификации и настроить их совместное применение с помощью логических операторов `И` и `ИЛИ`. Например, выражение `SYSTEM=DCE OR compat` указывает, что пользователь может войти в систему только в том случае, если будет успешно выполнена идентификация `DCE` или локальная идентификация (crypt()). Порядок элементов выражения в данном случае имеет значение.

Аналогичным образом в атрибуте **SYSTEM** можно указывать имена модулей идентификации. Например, если для атрибута **SYSTEM** указано выражение `SYSTEM=KRB5files OR compat`, хост AIX в первую очередь попытается выполнить идентификацию с помощью потока Kerberos, а затем, в случае неудачи, стандартную идентификацию AIX.

Атрибуты **SYSTEM** и **registry** всегда хранятся в файле `/etc/security/user` в локальной файловой системе. Если пользователь AIX определен в каталоге LDAP и для него заданы соответствующие атрибуты **SYSTEM** и **registry**, то в файле `/etc/security/user` будет содержаться запись для этого пользователя.

Атрибуты **SYSTEM** и **registry** пользователя можно изменить с помощь команды **chuser**.

Допустимые значения атрибута **SYSTEM** можно определить в файле `/usr/lib/security/methods.cfg`.

Примечание: Учетная запись root всегда идентифицируется с помощью локальных файлов защиты. Значение атрибута **SYSTEM** для пользователя root в файле `/etc/security/user` всегда равно `SYSTEM=compat`.

Для альтернативных способов идентификации применяется атрибут **SYSTEM**, который указывается в файле `/etc/security/user`. Например, распределенная вычислительная среда (DCE) запрашивает пароль, но проверяет его с помощью средств шифрования, отличных от применяемых в `/etc/passwd` и `/etc/security/passwd`. В разделах файла `/etc/security/user` для пользователей, идентифицируемых средствами DCE, можно указать `SYSTEM=DCE.`

Другие допустимые значения атрибута **SYSTEM** - это **compat**, **files** и **NONE**. Значение `compat` применяется в тех случаях, когда для определения имен (и последующей идентификации) применяется локальная база данных, а затем - база данных NIS. Значение `files` указывает, что будут применяться только локальные файлы. Значение `NONE` отключает все способы идентификации. Для полного отключения идентификации укажите значение `NONE` в строках **SYSTEM** и **auth1** раздела пользователя.

Другие допустимые значения для атрибута **SYSTEM** можно определить в файле `/usr/lib/security/methods.cfg`.

Примечание: Учетная запись root всегда идентифицируется с помощью локальных файлов защиты. Атрибут **SYSTEM** для пользователя root в файле `/etc/security/user` всегда равен `SYSTEM = "compat"`.

Дополнительная информация о парольной защите приведена в разделе *Управление операционной системой и устройствами*.

ИД пользователей

Во всех событиях контроля в протоколе указаны ИД пользователей, с помощью которых вы можете отслеживать их действия. Дополнительная информация об ИД пользователей приведена в разделе *Управление операционной системой и устройствами*.

Атрибуты пользователей и групп, поддерживаемые модулями идентификации

Для идентификации в AIX используется ряд атрибутов пользователей и групп.

В следующих таблицах приведен список большинства атрибутов пользователей и групп, а также указано, поддерживаются ли они различными загрузочными модулями. Каждая строка соответствует атрибуту, а каждый столбец - загрузочному модулю. Для атрибутов, поддерживаемых загрузочным модулем в столбце загрузочного модуля указано Да.

Примечание: Модули PKI и Kerberos используются только для идентификации, их необходимо совмещать с модулями базы данных (например LOCAL или LDAP). Они поддерживают дополнительные (расширенные) атрибуты и не поддерживают атрибуты, предоставленные в LOCAL или LDAP. Только для расширенных атрибутов отображаются пометки, даже если они аналогичны атрибутам LOCAL или LDAP.

Таблица 7. Атрибуты пользователей и поддержка модуля идентификации

Атрибут пользователя	Локальная база данных	NIS	LDAP	PKI	Kerberos
account_locked	Да	Нет	Да	Нет	Нет
admgroups	Да	Нет	Да	Нет	Нет
admin	Да	Нет	Да	Нет	Нет
auditclasses	Да	Нет	Да	Нет	Нет
auth_cert	Нет	Нет	Нет	Да	Нет
auth_domain	Да	Нет	Да	Нет	Нет
auth_name	Да	Нет	Да	Нет	Нет

Таблица 7. Атрибуты пользователей и поддержка модуля идентификации (продолжение)

Атрибут пользователя	Локальная база данных	NIS	LDAP	PKI	Kerberos
auth1 **Примечание:** Атрибут **auth1** устарел, и использовать его не рекомендуется.	Да	Нет	Да	Нет	Нет
auth2 **Примечание:** Атрибут **auth2** устарел, и использовать его не рекомендуется.	Да	Нет	Да	Нет	Нет
capabilities	Да	Нет	Да	Нет	Нет
core	Да	Нет	Да	Нет	Нет
core_compress	Да	Нет	Нет	Нет	Нет
core_hard	Да	Нет	Да	Нет	Нет
core_naming	Да	Нет	Нет	Нет	Нет
core_path	Да	Нет	Нет	Нет	Нет
core_pathname	Да	Нет	Нет	Нет	Нет
cpu	Да	Нет	Да	Нет	Нет
daemon	Да	Нет	Да	Нет	Нет
data	Да	Нет	Да	Нет	Нет
data_hard	Да	Нет	Да	Нет	Нет
dce_export	Да	Нет	Да	Нет	Нет
dictionlist	Да	Нет	Да	Нет	Нет
expires	Да	Нет	Да	Нет	Да
flags	Да	Нет	Да	Нет	Да
fsize	Да	Нет	Да	Нет	Нет
fsize_hard	Да	Нет	Да	Нет	Нет
funcmode	Да	Нет	Да	Нет	Нет
gecos	Да	Да	Да	Нет	Нет
groups	Да	Да	Да	Нет	Нет
groupsids	Да	Да	Да	Нет	Нет
histexpire	Да	Нет	Да	Нет	Нет
home	Да	Да	Да	Нет	Нет
host_last_login	Да	Нет	Да	Нет	Нет
host_last_unsuccessful_login	Да	Да	Да	Нет	Нет
hostsallowedlogin	Да	Нет	Да	Нет	Нет
hostsdeniedlogin	Да	Нет	Да	Нет	Нет
id	Да	Да	Да	Нет	Нет
krb5_attributes	Нет	Нет	Нет	Нет	Да
krb5_kvno	Нет	Нет	Нет	Нет	Да
krb5_last_pwd_change	Нет	Нет	Нет	Нет	Да
krb5_max_renewable_life	Нет	Нет	Нет	Нет	Да
krb5_mknvo	Нет	Нет	Нет	Нет	Да
krb5_mod_date	Нет	Нет	Нет	Нет	Да
krb5_mod_name	Нет	Нет	Нет	Нет	Да
krb5_names	Нет	Нет	Нет	Нет	Да
krb5_principal	Нет	Нет	Нет	Нет	Да
krb5_principal_name	Нет	Нет	Нет	Нет	Да
krb5_realm	Нет	Нет	Нет	Нет	Да

Таблица 7. Атрибуты пользователей и поддержка модуля идентификации (продолжение)

Атрибут пользователя	Локальная база данных	NIS	LDAP	PKI	Kerberos
lastupdate	Да	Да	Да	Нет	Нет
login	Да	Нет	Да	Нет	Нет
loginretries	Да	Нет	Да	Нет	Нет
logintimes	Да	Нет	Да	Нет	Нет
maxage	Да	Да	Да	Нет	Да
maxexpired	Да	Да	Да	Нет	Нет
maxrepeats	Да	Нет	Да	Нет	Нет
maxulogs	Да	Нет	Да	Нет	Нет
minage	Да	Да	Да	Нет	Нет
minalpha	Да	Нет	Да	Нет	Нет
mindiff	Да	Нет	Да	Нет	Нет
mindigit	Да	Нет	Да	Нет	Нет
minlen	Да	Нет	Да	Нет	Нет
minloweralpha	Да	Нет	Да	Нет	Нет
minother	Да	Нет	Да	Нет	Нет
minspecialchar	Да	Нет	Да	Нет	Нет
minupperalpha	Да	Нет	Да	Нет	Нет
nofiles	Да	Нет	Да	Нет	Нет
nofiles_hard	Да	Нет	Да	Нет	Нет
password	Да	Да	Да	Нет	Нет
pgid	Да	Да	Нет	Нет	Нет
pgrp	Да	Да	Да	Нет	Нет
projects	Да	Нет	Да	Нет	Нет
pwdchecks	Да	Нет	Да	Нет	Нет
pwdwarntime	Да	Нет	Да	Нет	Нет
rcmds	Да	Нет	Да	Нет	Нет
registry	Да	Нет	Нет	Нет	Нет
rlogin	Да	Нет	Да	Нет	Нет
roles	Да	Нет	Да	Нет	Нет
rss	Да	Нет	Да	Нет	Нет
rss_hard	Да	Нет	Да	Нет	Нет
screens	Да	Нет	Да	Нет	Нет
shell	Да	Да	Да	Нет	Нет
spassword	Да	Да	Да	Нет	Нет
stack	Да	Нет	Да	Нет	Нет
stack_hard	Да	Нет	Да	Нет	Нет
su	Да	Нет	Да	Нет	Нет
sugroups	Да	Нет	Да	Нет	Нет
sysenv	Да	Нет	Да	Нет	Нет
SYSTEM	Да	Нет	Нет	Нет	Нет
time_last_login	Да	Нет	Да	Нет	Нет
time_last_unsuccessful_login	Да	Нет	Да	Нет	Нет
tpath	Да	Нет	Да	Нет	Нет
tty_last_login	Да	Нет	Да	Нет	Нет

Таблица 7. Атрибуты пользователей и поддержка модуля идентификации (продолжение)

Атрибут пользователя	Локальная база данных	NIS	LDAP	PKI	Kerberos
tty_last_unsuccessful_login	Да	Нет	Да	Нет	Нет
ttys	Да	Нет	Да	Нет	Нет
umask	Да	Нет	Да	Нет	Нет
unsuccessful_login_count	Да	Нет	Да	Нет	Нет
unsuccessful_login_times	Да	Нет	Да	Нет	Нет
usrenv	Да	Нет	Да	Нет	Нет

Таблица 8. Атрибуты групп и поддержка модуля идентификации

Атрибут пользователя	Локальная база данных	NIS	LDAP	PKI	Kerberos
admin	Да	Нет	Да	Нет	Нет
adms	Да	Нет	Да	Нет	Нет
dce_export	Да	Нет	Да	Нет	Нет
id	Да	Да	Да	Нет	Нет
primary	Да	Нет	Да	Нет	Нет
projects	Да	Нет	Да	Нет	Нет
screens	Да	Нет	Да	Нет	Нет
users	Да	Да	Да	Нет	Нет

Система дисковых квот - обзор

Система дисковых квот позволяет системным администраторам управлять количеством файлов и блоков данных, выделяемых пользователям и группам.

Система дисковых квот - общие сведения:

Система дисковых квот, основанная на Berkeley Disk Quota System, является эффективным средством управления использованием дискового пространства. Систему квот можно определить для отдельных пользователей или групп пользователей. Она поддерживается каждой журнализированной файловой системой (JFS и JFS2).

Система дисковых квот устанавливает ограничения на основе следующих параметров, которые можно изменить с помощью команды **edquota** (для файловых систем JFS) или команды**j2edlimit** (для файловых систем JFS2):

- Гибкие ограничения для пользователя или группы
- Жесткие ограничения для пользователя или группы
- Период отсрочки квоты

Гибкое ограничение задает количество блоков диска объемом 1 Кб или файлов, использование которых будет разрешено пользователю или группе в обычных условиях работы. *Жесткое ограничение* задает максимальное количество блоков дискового пространства или файлов, которые могут быть зарезервированы или созданы пользователем в рамках действующих дисковых квот. *Период отсрочки квоты* позволяет пользователю на некоторое время (по умолчанию - одна неделя) превысить гибкое ограничение. Если за указанное время пользователь не сократит объем занимаемого им дискового пространства до установленного гибкого ограничения, то это ограничение будет зафиксировано для пользователя, и ему не будет выделено дополнительное пространство. Для восстановления прежнего состояния пользователь может удалить часть файлов, сделав общий объем используемого дискового пространства ниже гибкого ограничения.

Система дисковых квот ведет учет квот для пользователей и групп в файлах `quota.user` и `quota.group`. Эти файлы хранятся в корневых каталогах файловых систем, для которых установлены квоты. Файлы создаются командами **quotacheck** и **edquota**, а просмотреть их можно с помощью команд quota.

Восстановление после превышения квоты:

В этом разделе описывается восстановление после превышения квоты путем сокращения используемого пространства файловой системы.

Сократить используемое пространство файловой системы после превышения квот можно следующими способами:

- Остановите текущий процесс, вызвавший превышение квоты, удалите лишние файлы, затем повторно запустите программу, в которой возник сбой.
- При работе в текстовом редакторе, например vi, введите Esc-последовательность оболочки, чтобы проверить пространство файлов, удалить лишние файлы и затем вернуться к редактируемому файлу. Кроме того, при работе в оболочке C или Korn можно приостановить работу редактора, нажав Ctrl-Z, ввести необходимые команды работы с файловой системой и затем вернуться в редактор с помощью команды **fg** (foreground).
- Временно сохраните файл в файловой системе, в которой не превышена квота, удалите лишние файлы, затем верните сохраненный файл в прежнюю файловую систему.

Настройка системы дисковых квот:

Как правило, настройка дисковых квот необходима только для файловых систем, содержащих файлы и домашние каталоги пользователей.

Применять систему дисковых квот имеет смысл в следующих случаях:

- Дисковое пространство системы ограничено.
- Необходимо повысить надежность защиты файловых систем.
- Объем используемого дискового пространства и число пользователей достаточно велики (пример - университеты).

Если эти условия не выполняются в вашей среде, то задавать ограничения на объем используемого дискового пространства с помощью системы дисковых квот не рекомендуется.

Система дисковых квот может применяться только с журнализированными файловыми системами.

Примечание: Не настраивайте систему дисковых квот для файловой системы `/tmp`.

Для настройки системы дисковых квот выполните следующие действия:

1. Войдите в систему с правами доступа root.
2. Определите, для каких файловых систем необходимо задать квоты.

   Примечание: Так как многие редакторы и системные утилиты создают временные файлы в файловой системе `/tmp`, то для нее задавать квоты не следует.
3. С помощью команды **chfs** добавьте атрибуты **userquota** и **groupquota** в файл `/etc/filesystems`. В следующем примере с помощью команды **chfs** устанавливаются квоты пользователей на файловую систему`/home`:

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

   Для установления квот пользователей и групп в файловой системе `/home` введите следующую команду:

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

   Соответствующая запись в файле `/etc/filesystems` выглядит следующим образом:

```
/home:
dev         = /dev/hd1
vfs         = jfs
log         = /dev/hd8
mount       = true
check       = true
quota       = userquota,groupquota
options     = rw
```

4. Кроме того, можно указать имена альтернативных файлов дисковой квоты. По умолчанию применяются файлы `quota.user` и `quota.group` , расположенные в корневых каталогах файловых систем, для которых установлены квоты. Для этих файлов квот можно указать другие имена и каталоги. Это делается с помощью атрибутов **userquota** и **groupquota** в файле `/etc/filesystems`.

   В следующем примере с помощью команды **chfs** устанавливаются квоты пользователей и групп для файловой системы `/home`, а также задаются имена файлов квот `myquota.user` и `myquota.group`:

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

   Соответствующая запись в файле `/etc/filesystems` выглядит следующим образом:

   ```
   /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. Смонтируйте указанные файловые системы, если они еще не смонтированы.
6. Задайте необходимые ограничения для каждого пользователя и группы. С помощью команды **edquota** задайте для каждого пользователя и группы гибкие и жесткие ограничения дискового пространства и максимальное количество файлов.

   Ниже приведен пример записи квоты для пользователя *davec*:

   ```
   Квоты для пользователя davec:
   /home: blocks in use: 30, limits (soft = 100, hard = 150)
                  inodes in use: 73, limits (soft = 200, hard = 250)
   ```

   Этот пользователь занимает 30 Кб дискового пространства из разрешенных ему 100 Кб. Из максимально разрешенных 200 файлов пользователь *davec* создал 73. Для этого пользователя заданы буферы размером 50 Кб и 50 файлов для временного хранения данных.

   Устанавливая дисковые квоты для нескольких пользователей, можно указать в команде **edquota** флаг **-p**, чтобы скопировать квоты одного пользователя для другого.

   Для копирования квот пользователя *davec* для пользователя *nanc* введите следующую команду:

   ```
   edquota -p davec nanc
   ```

7. Включите систему квот с помощью команды **quotaon**. Команда**quotaon** активизирует квоты в указанной файловой системе или, если в командной строке указан флаг**-a**, то во всех файловых системах, для которых установлены квоты (согласно параметрам в файле `/etc/filesystems`).
8. С помощью команды **quotacheck** проверьте соответствие файлов квот фактическому уровню использования дискового пространства.

   Примечание: Эту операцию рекомендуется выполнять при первой активизации квот и после каждой загрузки системы. Команда **quotacheck** выполняется дольше в файловой системе JFS, чем в файловой системе JFS2 того же размера. Если квоты всегда устанавливаются до перезагрузки, то выполнять команду **quotacheck** в файловой системе во время перезагрузки не нужно.

   Для того чтобы проверка и активизация квот выполнялись во время загрузки системы, добавьте в конец файла `/etc/rc` следующие строки:

```
echo " Включение квот файловой системы "
/usr/sbin/quotacheck -a
/usr/sbin/quotaon -a
```

Разрешенное количество групп

Можно настроить и получить параметр разрешенного количества групп для AIX 7.1. Он определяет количество групп, в которые могут входить пользователи.

Разрешенное количество групп по умолчанию равно 128. Допустимы значения от 128 до 2048. Разрешенное количество групп задается в системном параметре конфигурации `v_ngroups_allowed` для устройства `sys0`. Значение параметра `v_ngroups_allowed` можно запросить у ядра или в базе данных ODM, и его можно изменить. При работе системы используется значение, заданное в ядре. Значение, заданное в базе данных ODM, применяется после перезагрузки системы.

Извлечение параметра разрешенного количества групп из базы данных ODM: Команды и процедуры позволяют извлечь параметр `v_ngroups_allowed`. Для извлечения параметра `v_ngroups_allowed` из базы данных ODM выполните команду **lsattr**.

Команда **lsattr** отображает параметр `v_ngroups_allowed` как атрибут `ngroups_allowed`. В следующем примере показано применение команды **lsattr** для извлечения атрибута `ngroups_allowed`:

```
$ lsattr -El sys0
SW_dist_intr    false           Включить рассылку SW прерываний                    True
autorestart     true            Автоматически перезагружать систему после сбоя     True
boottype        disk            н/д                                                False
capacity_inc    1.00            Приращение мощности процессора                     False
capped          true            Раздел с ограничениями                             False
conslogin       enable          Вход в систему с консоли                           False
cpuguard        enable          Предохранитель CPU                                 True
dedicated       true            Раздел выделенный                                  False
ent_capacity    4.00            Предписанная мощность процессора                   False
frequency       93750000        Тактовая частота системной шины                    False
fullcore        false           Включить полный дамп ядра                          True
fwversion       IBM,SPH01316    Версия встроенного ПО и уровни обновления          False
iostat          false           Постоянное ведение хронологии ввода-вывода диска   True
keylock         normal          Состояние системного замка во время загрузки       False
max_capacity    4.00            Максимальная возможная мощность процессора         False
max_logname     20              Максимальная длина имени входа во время загрузки   True
maxbuf          20              Макс. число страниц в кэше буфера блочного в/в     True
maxmbuf         0               Макс. число Кбайт фактической памяти для MBUFS     True
maxpout         0               Верхняя отметка для ожидающих запросов в/в файла   True
maxuproc        128             Максимальное число процессов пользователя          True
min_capacity    1.00            Минимальная возможная мощность процессора          False
minpout         0               Нижняя отметка для ожидающих запросов в/в файла    True
modelname       IBM,7044-270    Имя компьютера                                     False
ncargs          6               Размер списка ARG/ENV в блоках по 4 Кб             True
pre430core      false           Применение дампа ядра стиля до версии 430          True
pre520tune      disable         Режим совместимости с настройкой до версии 520     True
realmem         3145728         Объем доступной физической памяти в Кбайтах        False
rtasversion     1               Версия RTAS открытого встроенного ПО               False
sec_flags       0               Флаги защиты                                       True
sed_config      select          Режим Отключить обработку стека (SED)              True
systemid        IBM,0110B5F5F   ИД аппаратного обеспечения системы                 False
variable_weight 0               Переменный вес мощности процессора                 False
ngroups_allowed 128             Разрешенное количество групп при загрузке          True
$
```

Извлечение параметра разрешенного количества групп из ядра: Для извлечения параметра `v_ngroups_allowed` из ядра выполните процедуру `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("Разрешенное количество групп = %d\n",
myvar.v.v_ngroups_allowed.value);
else
printf("Процедура sys_parm() не выполнена rc = %d, errno = %d\n", rc, errno);
}
```

Изменение параметра разрешенного количества групп, заданного в базе данных ODM: Во время начальной загрузки системы необходимо задать значение разрешенного количества групп в ядре. Для изменения значения в базе данных ODM воспользуйтесь командой **chdev**. Это изменение вступит в силу после перезагрузки системы.

Для того чтобы изменить параметр **v_ngroups_allowed** в базе данных ODM с помощью команды **chdev**, введите:

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

Управление доступом на основе ролей

Администрирование системы является важным аспектом повседневной работы, а защита - неотъемлемой частью большинства функций системного администрирования. В дополнение к защите рабочей среды необходимо тщательно контролировать текущую работу системы.

Большинство сред требует, чтобы различные пользователи управляли различными обязанностями системного администрирования. Необходимо поддерживать разделение этих обязанностей таким образом, чтобы ни один пользователь, управляющий системой, не мог случайно или злоумышленно обойти защиту системы. В то время как традиционное системное администрирование UNIX не может достичь этих целей, они могут быть достигнуты с помощью управления доступом на основе ролей (RBAC).

Традиционные ограничения на администрирование UNIX

RBAC устраняет некоторые традиционные ограничения на администрирование системы UNIX. Они перечислены ниже:

Административная учетная запись root

Традиционно в AIX и других операционных системах UNIX определяется единственная учетная запись системного администратора **root** (обычно обозначаемая UID 0), которая может выполнять все привилегированные задачи администрирования системы. Предоставление права на выполнение всех административных задач единственному пользователю становится проблемой в случае разделения обязанностей. Хотя единая административная учетная запись приемлема в некоторых средах, многим средам требуется несколько администраторов, каждый из которых отвечает за свой, отличный от других набор административных задач.

Для того чтобы разделить административные обязанности между несколькими пользователями системы, ранее применялись такие приемы, как совместное использование пароля пользователя root или создание другого пользователя с тем же UID в качестве пользователя root. Этот способ разделения административных обязанностей небезупречен с точки зрения защиты, поскольку каждый администратор обладает полным контролем над системой, и нет никакого способа ограничить набор доступных ему операций. Так как пользователь root является наиболее привилегированным, пользователи root может выполнять несанкционированные операции, а также уничтожать все следы этих операций, делая невозможным их отслеживание.

Эскалация привилегий через SUID

Управление доступом в операционных системах UNIX ранее выполнялось с помощью UID, связанного с процессом. Однако UID root, равному 0, обычно разрешалось обходить проверки прав доступа. Таким образом, процесс, выполняемый под управлением пользователя root, может проходить любые проверки доступа и выполнять любые операции. Это серьезная проблема безопасности для концепции приложений **setuid** в UNIX.

Концепция **setuid** разрешает выполнять команду под другим идентификатором, чем тот, под управлением которого она была запущена. Это необходимо, когда обычному пользователю необходимо выполнить привилегированную задачу. Примером этого может служить команда AIX **passwd**. Поскольку у нормального пользователя нет доступа к файлу, в котором хранятся пароли пользователей, для изменения пароля пользователя необходимы дополнительные привилегии, так что команда **passwd** есть **setuid** для пользователя root. Когда обычный пользователь запускает команду **passwd**, с точки зрения операционной системы это выглядит как обращение пользователя root к файлу и предоставление такого доступа.

Хотя эта концепция обеспечивает необходимые функции, с ее применением связан неотъемлемый риск. Поскольку программа **setuid** фактически выполняется в контексте root, то в случае, если взломщик успешно перехватывает управление программой до ее завершения, он получает все права root и может обойти все проверки операционной системы и выполнять любые операции. Наилучшее решение заключается в том, чтобы присвоить программе подмножество привилегий пользователя root, так что соблюдается “Принцип минимальных прав доступа” на стр. 83 и угроза устраняется.

Элементы RBAC

RBAC позволяет создавать роли для администрирования системы и делегирования задач по администрированию какому-либо из группы защищенных пользователей системы. В AIX RBAC предоставляет механизм, с помощью которого функции администрирования, обычно предоставляемые только корневому пользователю, можно было назначить обычному пользователю системы.

RBAC реализует это путем определения функциональных обязанностей (ролей) в организации и назначения этих ролей определенным пользователям. RBAC представляет собой среду, которая позволяет администрировать систему путем использования ролей. Обычно роли определяются вместе с полномочиями на управление одним или несколькими административными аспектами среды. При назначении роли пользователю предоставляется набор разрешений или прав. Например, одна из управляющих ролей может быть предназначена для управления файловыми системами, а другая - для того чтобы обеспечить создание учетных записей пользователей.

По сравнению с традиционным администрированием UNIX RBAC имеет следующие преимущества:

- Администрировать систему могут различные пользователи, не имея при этом общего доступа к учетной записи.
- Изоляция защиты с помощью дискретного администрирования, поскольку каждый администратор получает только те полномочия, которые требуются.
- Позволяет принудительно применить принцип минимальных прав доступа. Пользователи и приложения получают только те права, которые необходимы, и только тогда, когда они требуются, благодаря чему снижается влияние на систему при возможной атаке.
- Позволяет согласованно реализовать и применить стратегии защиты в отношении управления системой и контроля доступа во всей компании.
- Определение роли можно создать один раз, а затем назначать его пользователям или удалять по необходимости при смене функциональных обязанностей пользователей.

Среда RBAC сфокусирована на следующих трех основных концепциях:

- Права доступа
- Роли
- Привилегии

В комбинации эти три концепции позволяют системе RBAC применять принцип минимальных прав доступа.

Права доступа:

Права доступа представляют собой строку текста, связанную с функциями или командами, которые имеют отношение к защите. Благодаря механизму предоставления прав доступа пользователи могут выполнять привилегированные действия, а различные классы пользователей имеют разные уровни функциональности.

При выполнении команды под управлением прав доступа вызвавший ее пользователь получает доступ только при условии, что он обладает соответствующими правами. Права доступа можно сравнить с ключом, который может открыть доступ к одной или нескольким командам. Права доступа не предоставляются непосредственно пользователям. Пользователям назначаются роли, которые представляют собой наборы прав доступа.

Роли:

Роли позволяют группировать наборы функций управления в системе. Если провести аналогию между правами доступа и ключом, то роль можно представить в виде связки ключей, на которой хранится несколько наборов прав доступа. Права доступа можно назначить для роли непосредственно или косвенно с использованием второстепенной роли. Второстепенная роль - это другая роль, от которой данная роль наследует права доступа.

Роль сама по себе не дает пользователю дополнительных полномочий, но при этом она служит механизмом сбора прав доступа и инструментом для назначения прав доступа пользователю. При определении роли и ее назначении пользователю определяются административные задачи, которые может выполнять пользователь. После определения роли администратор ролей может назначить ее одному или нескольким пользователям для выполнения привилегированных операций, представляемых ролью. Кроме того, одному пользователю можно назначить несколько ролей. После того, как пользователю назначена роль, он может использовать права доступа, которые предоставлены роли, для разблокирования доступа к административным командам системы.

Организационные стратегии и процедуры определяют, каким образом роли распределяются между пользователями. Не следует предоставлять роли слишком много прав и назначать одну роль большому количеству пользователей. Большинство ролей должны назначаться только для руководящего персонала. Поскольку права пользователя root исторически предоставляются только надежным пользователям, роли также следует назначать лишь надежным пользователям. Предоставляйте роли только тем пользователям, чьи потребности в них обоснованы, и только на требуемый период. Это сократит шансы незарегистрированных пользователей на получение прав и злоупотребление ими.

Привилегии:

Привилегия - это атрибут процесса, позволяющий ему обойти системные ограничения и запреты.

Механизм привилегий предоставляет надежным приложениям средства, недоступные прочим приложениям. Например, с помощью привилегий можно переопределить ограничения защиты, разрешить расширенное использование определенных системных ресурсов, таких как память и дисковое пространство, и скорректировать производительность и приоритет процесса. Привилегию можно понимать как средство, позволяющее процессу обойти конкретное ограничение защиты в системе.

Права доступа и роли - это инструменты уровня пользователя, позволяющие пользователю получать доступ к привилегированным операциям. С другой стороны, привилегии - это механизм ограничений, используемый в ядре с целью определить, разрешено ли процессу выполнять конкретное действие.

Привилегии связаны с процессом и обычно приобретаются путем выполнения привилегированной команды. Благодаря этим связанным привилегиям процесс может выполнять соответствующие привилегированные

операции. Например, если пользователь применяет роль, у которой есть права на выполнение команды, то при запуске команды процессу присваивается набор привилегий.

Принцип минимальных прав доступа:

Некоторые операции в операционной системе требуют наличия прав на их выполнение, поэтому выполнять их могут только пользователи с такими правами. В число этих операций часто входят такие задания, как перезагрузка системы, добавление и изменение файловых систем, добавление и удаление пользователей и изменение системных даты и времени.

В традиционных системах UNIX процесс или пользователь может находиться в обычном или в привилегированном режиме (который также называется администратором или корневым). Процесс, исполняемый как root может выполнить любую команду или системную операцию, а обычный пользователь не может выполнять привилегированные операции. Традиционные системы UNIX реализуют концепцию предоставления либо всех прав, либо никаких, и не защищены от действий администратора со слишком большим количеством прав доступа.

Традиционный подход UNIX, согласно которому в единственном привилегированном режиме предоставляется неограниченный доступ к системе, слишком груб для того, чтобы удовлетворять требования систем с высокой степенью защиты. Чтобы система была защищенной, каждому процессу следует предоставлять как можно более ограниченное число прав доступа, необходимых для выполнения задачи. Благодаря использованию прав доступа только тот процесс, для которого они необходимы, получит эти права. Такой подход к ограничению прав называется принципом минимальных прав доступа и служит для снижения числа повреждений системы из-за небрежных или злонамеренных действий администраторов и операторов.

Например, для изменения пароля нужны определенные права доступа к файлам, и обычный пользователь чаще всего не обладает такими правами. Если бы у пользователей всегда были такие права, то они могли бы совершать действия, которые нежелательны с точки зрения безопасности. Поэтому необходимые права доступа предоставляются только команде **passwd**, а не всем пользователям.

В среде RBAC пользователи сами по себе изначально не имеют прав доступа. Пользователям разрешено запускать определенные команды, которым и предоставляются права. Если бы вместо этого права доступа были предоставлены непосредственно пользователю, то он мог бы использовать их когда угодно и каким угодно образом. Предоставление прав доступа позволяет ограничить контекст, в котором они применяются. Это приводит к усилению защиты, поскольку когда взломщик использует защищенное приложение, он обладает ограниченным числом прав доступа, а не полными правами корневого пользователя.

Перед тем, как предоставить права доступа защищенному приложению, следует тщательно проверить его. Кроме того, права доступа должны предоставляться в то время и в тех случаях, когда это необходимо для приложения. Защищенные приложения - это такие же программы, как и остальные. Единственная разница заключается в том, что защищенным приложениям предоставляются права на выполнение действий, которые запрещено выполнять незащищенным приложениям.

RBAC AIX

AIX предоставляла ограниченную реализацию RBAC перед версией AIX 6.1.

С системы AIX 6.1 началось внедрение реализации RBAC, которая обеспечивает высокую дискретность задач системного администрирования. Поскольку эти реализации RBAC очень различны по функциональности, будут использоваться следующие термины:

Обычный режим RBAC
: Прежнее поведение ролей AIX, относящихся к версиям до AIX 6.1

Расширенный режим RBAC
: Новая реализация, внедренная начиная с AIX 6.1

Поддерживаются оба режима работы. При этом, расширенный режим RBAC устанавливается в системе AIX 6.1 по умолчанию. В следующих разделах приведено краткое описание обоих режимов и различий между ними, а также информация о настройке системы для работы в требуемом режиме RBAC.

Устаревший режим RBAC:

До AIX 6.1 в AIX предоставлялся ограниченный набор функций RBAC, позволявший пользователям, отличным от root, выполнять определенные задачи администрирования системы.

В данной реализации RBAC, когда пользователь, отличный от root, запускает заданную административную команду, код в этой команде определяет, присвоена ли пользователю роль с необходимыми правами доступа. Если да, то выполнение команды продолжается. Если нет, то происходит сбой и выдается ошибка команды. Часто требуется, чтобы команда, контролируемая правами доступа, была **setuid** для пользователя root, чтобы у уполномоченного инициатора запуска были привилегии, необходимые для выполнения операции.

В данной реализации RBAC также введен предопределенный, но расширяемый пользователем набор прав доступа, с помощью которых можно определять доступ к административным командам. Кроме того, предоставляется структура административных команд и интерфейсов, предназначенная для создания ролей, присвоения прав доступа ролям и присвоения ролей пользователям.

Хотя данная реализация позволяет частично сегментировать ответственность за администрирование системы, она действует со следующими ограничениями:

1. Изменения в командах и приложениях должны поддерживать RBAC.
2. Предопределенные права доступа не являются детализированными, а механизмы создания прав доступа - устойчивыми.
3. Для запуска команды часто требуется членство в определенной группе, а также наличие роли с заданными правами доступа.
4. Разделение обязанностей труднореализуемо. Если пользователю присвоено несколько ролей, то нет никакого способа действовать только с одной ролью. У пользователя есть все права доступа всех его ролей.
5. Принцип наименьших привилегий не применяется в операционной системе. Команды, как правило, должны быть SUID для пользователя root.

Устаревший режим RBAC поддерживается в целях обеспечения совместимости, но по умолчанию применяется расширенный режим RBAC. Расширенный режим RBAC предпочтителен в AIX.

Расширенный режим RBAC:

С AIX 6.1 поставляется более мощная реализация RBAC. Приложения, которым для определенных операций необходимы права администратора, имеют новые опции интеграции в расширенной инфраструктуре RBAC в AIX.

Эти опции интеграции сфокусированы на использовании дискретных прав доступа и способности сделать любую команду системы привилегированной командой. Функции расширенного режима RBAC устанавливаются и включаются по умолчанию во всех вариантах установки AIX, начиная с AIX 6.1.

Расширенный режим RBAC предоставляет настраиваемый набор прав доступа, ролей, привилегированных команд, устройств и файлов с помощью перечисленных ниже баз данных RBAC. При использовании расширенного режима RBAC можно либо расположить базы данных в локальной системе, либо управлять ими удаленно с помощью LDAP.

- База данных прав доступа
- База данных ролей
- База данных привилегированных операторов

- База данных привилегированных устройств
- База данных привилегированных файлов

В расширенном режиме RBAC реализованы новые соглашения об именовании прав, благодаря чему можно создавать их иерархии. AIX предоставляет дискретный набор системных прав доступа, а администратор может создавать дополнительные пользовательские права доступа по своему усмотрению.

Поведение ролей усовершенствовано с точки зрения разделения функциональных обязанностей. В расширенном режиме RBAC реализована концепция сеансов ролей. Сеансом роли является процесс, с которым связана одна или несколько ролей. Таким образом, пользователь может создавать сеансы для любых назначенных ему ролей, активируя отдельную роль или несколько ролей одновременно. По умолчанию с новым системным процессом не связана ни одна роль. Кроме того, усовершенствованные роли удовлетворяют требованиям относительно идентификации пользователя перед активацией роли. Таким образом, пользовательские роли защищены от атак, поскольку теперь для активации ролей пользователя атакующий должен идентифицироваться.

Внедрение базы данных привилегированных команд реализует принцип минимальных прав доступа. Увеличена дискретность системных прав доступа, благодаря чему можно явно указывать права доступа для команды, и права доступа будут управлять выполнением команды. Этим обеспечивается возможность проводить принудительные проверки прав доступа на выполнение команд, не требуя изменять исходный код команды. С использованием баз данных привилегированных команд исчезает необходимость в приложениях SUID и SGID, поскольку могут быть предоставлены только требуемые права доступа.

База данных привилегированных устройств позволяет управлять доступом к устройствам с помощью прав доступа, а база данных привилегированных файлов позволяет обычным пользователям обращаться к важным файлам на основании прав доступа. Эти базы данных увеличивают дискретность административных задач системы, которые могут быть назначены пользователям, не имеющим остальных прав доступа.

Собранная в базах данных RBAC информация проверяется, а затем отправляется в область ядра, называемую таблицами защиты ядра (KST). Необходимо отметить, что состояние данных в KST определяет стратегию защиты системы. Записи, которые изменены в базах данных RBAC на пользовательском уровне, не используются для принятия решений в отношении защиты, пока эта информация не будет отправлена в KST с помощью команды **setkst**.

Настройка режима RBAC:

Режимом RBAC можно управлять с помощью системной конфигурационной переменной ядра. Эта переменная указывает, включен или отключен расширенный режим RBAC.

В AIX 6.1 и более поздних версий расширенный режим RBAC включен по умолчанию. Для отключения расширенного режима RBAC и возврата к обычному режиму RBAC можно запустить команду **chdev** для устройства **sys0** и присвоить атрибуту **enhanced_RBAC** значение **false**. Для того чтобы изменение атрибута **enhanced_RBAC** вступило в силу, перезагрузите систему. Для включения расширенного режима RBAC атрибут **enhanced_RBAC** должен иметь значение **true**. Задать или прочесть режим можно и программно - с помощью системного вызова **sys_parm()**.

Для чтения текущего состояния RBAC выполните в системе следующую команду:

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

После этого можно отключить расширенный режим RBAC с помощью следующей команды, а затем перезагрузить систему:

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

В среде WPAR режим RBAC можно настроить только из глобальной системы, и такие настройки затронут как глобальную систему, так и все WPAR в этой системе.

Сравнение устаревшего режима RBAC и расширенного режима RBAC:

Существующие и новые интерфейсы изменились: теперь они проверяют конфигурацию системы и либо запускают новый код, либо действуют по-старому.

В устаревшем режиме RBAC применяются только права доступа, проверяемые в коде самой команды. Таблицы защиты ядра (KST) никак не влияют ни на выполнение команд, ни на проверки прав доступа. Выяснение наличия у пользователя прав доступа происходит по правилам устаревшего режима RBAC: все права доступа пользователя извлекаются и проверяются на предмет соответствия. Новые возможности, такие как команда **swrole** и атрибуты **default_roles** и **auth_mode**, в устаревшем режиме RBAC недоступны. Однако новые привилегии, права доступа и команды управления правами доступа поддерживаются в устаревшем режиме RBAC.

В следующей таблице перечислены некоторые различия между устаревшим и расширенным режимами RBAC.

Таблица 9. различия между устаревшим и расширенным режимами RBAC

Компонент	Устаревший RBAC	Расширенный RBAC
Активация роли	Все роли пользователя всегда активны	По умолчанию роли не активны, пока не будут активизированы явно командой **swrole**
Атрибут **default_roles**	Недоступна	Поддерживается
Команда **swrole**	Недоступна	Поддерживается
Команды управления ролями	Поддерживается	Поддерживается
Команды управления правами доступа	Поддерживается	Поддерживается
Иерархия прав доступа	Все права доступа независимы друг от друга. Иерархия отсутствует.	Поддерживается концепция иерархии прав доступа, когда одни права доступа могут быть родительскими по отношению к другим правам доступа
Проверки прав доступа	Проводятся, только если команда сама проверяет права доступа	Проводятся Базой данных привилегированных команд и/или самой командой
Детализированные привилегии	Поддерживается	Поддерживается
Команда **pvi**	Недоступна	Поддерживается
Таблицы защиты ядра	Недоступна	Поддерживается
Расположение базы данных RBAC	Локальные файлы	Локальные файлы или LDAP

Применение расширенного RBAC

Для успешного применения расширенного RBAC системные администраторы должны хорошо разбираться в следующих областях.

Права доступа RBAC:

Права доступа играют важную роль в Ролевом управлении доступом (RBAC). На основе строк прав доступа операционная система определяет, разрешено ли выполнять привилегированную операцию. Связанные проверки могут выполняться явно - в коде, либо загрузчиком - во время выполнения защищенных привилегированных исполняемых файлов.

Имена строк прав доступа обозначают привилегированные операции, которые они представляют и контролируют. Соглашение об именах прав доступа AIX поддерживает иерархическую структуру, задаваемую текстовым именем прав доступа. Строки прав доступа AIX задаются в формате с точками, описывающем иерархию прав доступа. Например, права на создание файловых систем имеют вид

aix.fs.manage.create. Если эти права доступа включены в роль, то пользователь, которому присвоена эта роль, может создавать файловые системы AIX. Если в роль включены родительские права доступа **aix.fs.manage**, то пользователь, которому присвоена эта роль, может, помимо создания файловых систем, выполнять и другие задачи управления файловыми системами.

RBAC AIX различает системные (предопределенные) и пользовательские (созданные после установки) права доступа.

Системные права доступа:

AIX предоставляет предопределенный и неизменяемый набор прав доступа. Они называются системными правами доступа. Эти права доступа связаны с различными привилегированными операциями AIX; соответствующая связь задается в базе данных привилегированных команд.

Наверху иерархии системных прав доступа находятся права доступа **aix**. Они являются родительскими для всех остальных системных прав доступа. Предоставление этих прав доступа роли означает предоставление ей всех системных прав доступа. Для просмотра полного набора системных прав доступа AIX и краткого описания каждого типа прав доступа выполните следующую команду:

```
lsauth -f -a description ALL_SYS
```

Вывод этой команды показывает, что список системных прав доступа - это многоуровневая иерархия. Например, у прав доступа **aix** есть несколько прямых потомков. Каждый из них является родителем прав доступа следующего уровня иерархии. Права доступа **aix.fs** включают несколько дочерних прав доступа, например **aix.fs.manage**, которые, в свою очередь, содержат права доступа **aix.fs.manage.change**, **aix.fs.manage.create** и др.

Пользовательские права доступа:

В дополнение к системным правам доступа, RBAC AIX позволяет системным администраторам определять собственные права доступа в базе данных прав доступа (/etc/security/authorizations). Они называются пользовательскими правами доступа.

Системный администратор может добавлять, изменять и удалять пользовательские права доступа. Например, системный администратор может разрешить некоторым пользователям выполнять привилегированную команду, создав пользовательские права доступа, связав их с командой и предоставив их роли, присвоенной таким пользователям.

Пользовательские права доступа функционируют в рамках той же иерархии, что и системные права доступа. Однако на имена пользовательских прав доступа AIX наложены некоторые ограничения.

- Пользовательские права доступа должны быть определены под новым предком верхнего уровня. Иными словами, пользовательские права доступа не могут быть потомками системных прав доступа (**aix**).
- Имя прав доступа может содержать до 63 печатаемых символов.
- Уровень вложенности прав доступа не может превышать восьми.
- У прав доступа может быть произвольное число прямых потомков, но только один прямой предок. Два независимых экземпляра прав доступа не могут иметь одного и того же прямого потомка.

Поскольку иерархия запрещает элементу иметь несколько прямых предков, вы не можете создать пользовательские права доступа, которые были бы предком существующих системных прав доступа. По этой причине, создать права доступа с именем **aix.custom** не удастся, а создание прав доступа с именем **custom.aix** приведет к появлению совершенно новых прав доступа, а не предка системных прав доступа **aix**.

При создании пользовательских прав доступа рекомендуется пользоваться следующим синтаксисом, во избежание конфликтов имен между различными компонентами программного обеспечения:

*имя_вендора***.***имя_продукта***.***функция***.***функция_1***.***функция_2*...

имя_вендора
Указывает имя вендора модуля программного обеспечения.

имя_продукта
Имя высокоуровневого продукта, контролируемого RBAC.

функция, *функция_1*, *функция_2* **...**
Эти строки представляют функции, контролируемые RBAC. Кроме того, эти строки предоставляют иерархию этих функций.

Например, **ibm.db2.manage** может представлять управленческие аспекты комплекта баз данных IBM DB2. Как указывалось ранее, *имя_вендора* **aix** зарезервировано для использования в AIX и не может применяться для пользовательских прав доступа.

Существует несколько команд управления правами доступа, с помощью которых системные администраторы могут просматривать, создавать, изменять и удалять пользовательские права доступа. Создавать пользовательские права доступа можно командой **mkauth**, изменять - командой **chauth**, удалять - командой **rmauth** и просматривать - командой **lsauth**. Для просмотра всех пользовательских прав доступа и краткого описания каждого типа прав доступа выполните следующую команду:

```
lsauth -f -a description ALL_USR
```

Перед созданием пользовательских прав доступа ответьте на следующие вопросы:

- Будет ли оправданным применять существующие системные права доступа вместо создания новых пользовательских прав доступа?
- Принадлежат ли новые права доступа существующей иерархии пользовательских прав доступа или это первые права доступа новой иерархии?
- Если это новая иерархия, то какова ее структура?
- Каково текстовое описание прав доступа?
- Требуется ли языковой перевод описания прав доступа?
- Имеет ли смысл указывать определенный ИД прав доступа при создании прав доступа? Рекомендуется создать ИД прав доступа командой **mkauth**.

Ответив на эту вопросы, выполните следующие действия для создания прав доступа:

1. Если требуется языковой перевод, создайте или добавьте описание в каталог сообщений.
2. С помощью команды **mkauth** создайте все родительские права доступа в иерархии, если они еще не существуют.
3. С помощью команды **mkauth** создайте требуемые права доступа. Если необходимо конкретное значение, укажите в команде атрибут **id**.

Перенос устаревших прав доступа:

До появления AIX версии 6.1 в операционной системе существовал ограниченный предопределенный набор прав доступа, которые распознавались операционной системой. Эти права доступа не определялись в каком-либо файле системы, а сразу присваивались ролям. Для того чтобы эти устаревшие права доступа поддерживались в новой структуре RBAC AIX версии 6.1 и более поздних версий, эти права определяются как пользовательские и предоставляются по умолчанию в базе данных прав доступа.

Поскольку операционная система AIX переходит на новое соглашение об именах прав доступа, все проверки старых имен прав доступа в операционной системе AIX модифицированы: теперь они дополнительно проверяют соответствующие новые права доступа и разрешают доступ только в том случае, если для процесса существуют оба типа прав доступа. В следующей таблице перечислены устаревшие предопределенные права доступа и соответствующие им новые системные права доступа.

Существующие права доступа AIX	Соответствующие новые права доступа
Backup	aix.fs.manage.backup
Diagnostics	aix.system.config.diag
DiskQuotaAdmin	aix.fs.manage.quota
GroupAdmin	aix.security.group
ListAuditClasses	aix.security.audit.list
PasswdAdmin	aix.security.passwd
PasswdManage	aix.security.passwd.normal
UserAdmin	aix.security.user
UserAudit	aix.security.user.change
RoleAdmin	aix.security.role
Restore	aix.fs.manage.restore

Роли RBAC:

Роли - это механизм присвоения прав доступа пользователю и группирования набора задач администрирования системы. Роль AIX - это преимущественно контейнер для набора прав доступа.

В AIX права доступа могут присваиваться роли напрямую или косвенно - через суброль. Указать суброль для роли можно в атрибуте **rolelist** роли. После настройки суброли для роли последняя получает все права доступа этой суброли.

Присвоение роли пользователю позволяет ему обращаться к роли и применять все содержащиеся в ней права доступа. Системный администратор может присвоить роль нескольким пользователям, а также присвоить несколько ролей одному пользователю. Пользователь, которому присвоено несколько ролей, может активировать несколько ролей (до восьми) одновременно, если это необходимо для выполнения функций управления системой.

AIX предоставляет набор предопределенных ролей, предназначенных для управления системой. Однако предполагается, что пользователи будут создавать свои собственные роли или изменять существующие предопределенные роли. Существуют команды, позволяющие просматривать, создавать, изменять и удалять роли AIX. Создавать роли можно командой **mkrole**, изменять - командой **chrole**, удалять - командой **rmrole** и просматривать - командой **lsrole**.

При создании новой роли AIX ответьте на следующие вопросы:

- Каково будет имя роли?
- Имя роли - это произвольная текстовая строка, однако рекомендуется, чтобы она давала некоторое представление о предназначении роли. Имя роли может содержать до 63 печатаемых символов.
- Какие права доступа необходимы для роли? Определите, следует ли присвоить роли права доступа напрямую или косвенно - через суброль.
- Нужно ли будет пользователю подтверждать свою подлинность при активации роли?

Активация роли:

По умолчанию в AIX версии 6.1 или более поздних версий с расширенным RBAC при идентификации пользователя в системе с сеансом пользователя не связаны никакие роли и права доступа. Для того чтобы связать роли с сеансом пользователю необходимо вызвать отдельную команду идентификации (команду **swrole**), с помощью которой можно сменять роли.

Пользователь может активировать только те роли, которые назначались ему прежде. По умолчанию от пользователя требуется идентификация только при входе в сеанс роли или при добавлении роли к сеансу. С помощью атрибута **auth_mode** можно при необходимости указать, что роль не должна требовать идентификации.

При переключении к новому сеансу роли создается оболочка (сеанс) без наследования ролей от предыдущего сеанса. Это осуществляется путем создания оболочки процесса для роли и назначения ему нового ИД роли (RID). Создание сеанса напоминает использование команды **su** с той разницей, что в этом случае изменяется только ИД роли процесса, а такие характеристики, как UID и GID, остаются неизменными. Команда **swrole** позволяет пользователю создать сеанс роли для одиночной роли или для нескольких ролей одновременно. Пользователь может переключаться из текущего сеанса роли в новый сеанс роли без каких-либо ограничений. Поскольку новый сеанс является новым процессом, этот новый сеанс не наследует роли от предыдущего. Для восстановления предыдущего сеанса пользователь должен выйти из текущего сеанса роли. Роли, предусмотренные сеансом (активный набор ролей) можно просмотреть, с помощью команды **rolelist**, запущенной из сеанса. Кроме того, администратор может использовать команду **rolelist** для просмотра набора активных ролей для данного системного процесса.

При необходимости пользователю можно назначить набор ролей по умолчанию с помощью нового атрибута **default_roles**. Этот атрибут предназначен для ситуаций, когда процессы, создаваемые от имени пользователя, всегда должны быть связаны с данным набором ролей, как, например, команда **cron**. Команда cron работает в фоновом режиме и выполняет команды как определенный пользователь. Возможно, некоторые выполняемые команды будут требовать идентификации. Для этого необходимо указать, что для ИД пользователя всегда должен быть активен набор ролей, поскольку команда **cron** не имеет механизма, который позволил бы ей получить эти роли впоследствии. В атрибуте **default_roles** можно указать до восьми имен ролей или присвоить ему особое значение **ALL**. Установка **default_roles=ALL** назначает для сеанса все роли пользователя. В случае, если пользователю назначено более восьми ролей, то для такого сеанса будут включены только первые восемь ролей.

Максимальное число ролей в сеансе:

В расширенном RBAC системный администратор может настроить на общесистемной основе максимальное число ролей, которые пользователь может активировать в заданном ролевом сеансе. По умолчанию пользователь может активировать до восьми ролей в сеансе.

В некоторых средах может потребоваться большее разделение обязанностей, при котором пользователю будет разрешено активировать только одну роль в каждый момент времени. В этих средах атрибут **maxroles** раздела **usw** файла `/etc/security/login.cfg` можно изменить, ограничив максимальное число ролей в сеансе. Атрибуту **maxroles** можно присвоить значение от 1 до 8, обозначив максимально допустимое число ролей в сеансе.

Для просмотра текущего значения ограничения на число ролей в сеансе выполните следующую команду:

```
lssec -f /etc/security/login.cfg -s usw -a maxroles
```

Для того чтобы изменить систему, разрешив пользователю активировать только одну роль в каждый момент времени, выполните следующую команду:

```
chsec -f /etc/security/login.cfg -s usw -a maxroles=1
```

Изменение атрибута **maxroles** вступает в силу немедленно для всех вновь создаваемых ролевых сеансов, не требуя перезагрузки системы. На ролевые сеансы, существовавшие до изменения значения, изменение не влияет. Максимальное число ролей в сеансе начинает действовать в момент инициализации сеанса.

Предопределенные роли:

В локальной базе данных ролей (`/etc/security/roles`) в новой версии AIX версии 6.1 и более поздних версиях поставляется предопределенный набор ролей. Он предназначен для группирования типичных административных обязанностей.

Этот набор ролей служит в качестве предлагаемого средства разделения административных обязанностей. Администраторы ролей могут изменять или удалять эти роли или создавать новые роли, в соответствии с потребностями среды. Ниже перечислены предоставляемые роли и дано краткое описание возможностей каждой роли.

Имя роли	**Описание роли**
auditadm	Администратор аудита. Роль auditadm отвечает за настройку контроля и ведения протоколов стратегий системы, включая атрибуты всей системы, отдельного пользователя и отдельной роли. Эта роль имеет права доступа для просмотра контрольного журнала.
fsadm	Администратор файловой системы. Роль fsadm создает файловые системы и делает их доступными для пользователей в системе. Ниже приведены некоторые функции роли fsadm: • Указание стратегий монтирования • Совместное использование стратегий • Присвоение квот • Определение уровня сжатия • Установление форматов файловых систем • Выполнение операций резервного копирования и восстановления
isso	Information System Security Officer. ISSO отвечает за создание и присвоение ролей и поэтому является самой могущественной ролью в системе. Ниже перечислены некоторые обязанности ISSO: • Создание и обслуживание стратегии защиты • Задание паролей для пользователей • Конфигурация сети • Администрирование устройств
pkgadm	Администратор пакетов программ. Роль pkgadm отвечает за программное обеспечение, установленное в системе, и по умолчанию имеет права доступа для установки, обновления и удаления системного программного обеспечения.
sa	System Administrator. Роль SA предоставляет функции каждодневного администрирования. Ее обязанности: • Администрирование пользователей (кроме задания пароля) • Администрирование файловой системы • Обновление программного обеспечения • Управление сетевыми демонами • Выделение устройств

Имя роли	Описание роли
secadm	Администратор защиты. Роль secadm обслуживает параметры защиты в системе. Эта роль присваивает пользователям такие атрибуты, как членство в группах, роли, права доступа и допуски, а также присваивает еще не назначенные роли. Эта роль также может присвоить атрибуты защиты объектам системы, включая параметры RBAC, списки прав доступа, принадлежность и членство. Ниже приведены некоторые функции роли secadm: • Назначение паролей новым учетным записям пользователей • Разблокирование заблокированных учетных записей
so	System Operator. Роль SO предоставляет функции каждодневной работы. Ее обязанности: • Завершение работы и перезагрузка системы • Резервное копирование, восстановление и выделение квот для файловой системы • Ведение протокола ошибок системы, трассировка и статистика • Администрирование рабочей нагрузки
svcadm	Администратор служб. Роль svcadm включает, настраивает и отключает системные службы. Эта роль позволяет настраивать сетевые атрибуты, такие как IP-адреса, маршруты, имена хостов и стратегии брандмауэров.
sysop	System Operator. Роль sysop обслуживает всю систему с правами доступа, разрешающими выполнять диагностику системы и обычное ее обслуживание. Некоторые задачи роли sysop приведены ниже: • Очистка файлов протоколов и очередей печати • Остановка и перезапуск систем
useradm	Администратор пользователей. Эта роль отвечает за задачи высшего уровня, связанные с обслуживанием пользователей без управления паролями. Роль useradm создает, изменяет и удаляет учетные записи пользователей, как определено стандартными параметрами защиты. Также эта роль может создавать дополнительные роли и группы со стандартными параметрами защиты.

Перенос ролей:

Если система AIX версии ниже AIX версии 6.1 обновляется до уровня расширенного RBAC AIX посредством установки с переносом, то при переносе файла `/etc/security/roles` происходит попытка добавить новые функции в файл, сохраняя текущие возможности роли.

Определения ролей в файле сохраняются и лишь слегка модифицируются: в них добавляется уникальный ИД роли, чтобы роль могла правильно функционировать в новой структуре. Любые права доступа в файле `/etc/security/roles`, отличные от предопределенных, считаются пользовательскими. Во время переноса имена этих прав доступа добавляются в качестве записей в локальную базу данных прав доступа `/etc/security/authorizations`. В дополнение к переносу старых определений ролей, в файл добавляются новые предопределенные роли. По окончании переноса системный администратор должен убедиться, что права доступа и роли правильно определены в среде.

Привилегии RBAC:

В структуре расширенного RBAC системные привилегии играют важную роль как средство, позволяющее непривилегированным пользователям выполнять привилегированные задачи. Привилегия - это механизм предоставления процессу дополнительных функций в системных вызовах.

Концепция привилегий в основном относится к уровню ядра, поскольку определение и большая часть проверки выполняются именно в ядре. Однако предоставляются и интерфейсы пользовательского уровня, предназначенные для обработки назначения привилегий командам, устройствам и процессам.

Важно отметить разницу между привилегиями и правами доступа. И те, и другие служат для контроля некоторых допустимых исключений из стратегии защиты системы. Принципиальное различие заключается в том, что привилегии связаны с конкретными процессами, а права доступа - с пользователями (посредством ролей). Права доступа находятся в роли и связаны с пользователем, имеющим эту роль; они не зависят от выполняемой программы. Привилегии находятся в программе и предоставляют механизм тонкой настройки стратегии защиты системы. Благодаря этим связанным привилегиям процесс может выполнять соответствующие привилегированные операции.

Привилегии определяются в ядре AIX как отдельные биты маски, обеспечивающие управление доступом к привилегированным операциям. В AIX предоставляется более 100 привилегий, что позволяет очень тонко настраивать контроль над привилегированными операциями. При определении доступа в системном вызове ядро выясняет, есть ли у процесса соответствующий бит привилегии, и затем принимает или отклоняет запрос.

Привилегии присваиваются вызовам команд через базу данных привилегированных команд и применяются для управления доступом к устройствам через базу данных привилегированных устройств.

Имена и иерархия привилегий:

Системный администратор AIX не может создавать, изменять или удалять привилегии.

Список имеющихся привилегий и краткое описание привилегий можно просмотреть с помощью следующей команды:

```
lspriv -v
```

Привилегии, предоставляемые в AIX, перечислены в привилегиях AIX . Текстовые имена всех привилегий AIX начинаются с **PV_**. Символы после префикса **PV_** обозначают иерархические взаимосвязи между привилегиями. Например, привилегия контроля **PV_AU_** является родительской для привилегий **PV_AU_ADD**, **PV_AU_ADMIN**, **PV_AU_READ**, **PV_AU_WRITE** и **PV_AU_PROC**. При выяснении привилегий система сначала проверяет, есть ли у процесса наименьшая необходимая привилегия, а затем движется вверх по иерархии, проверяя наличие более широких привилегий. Привилегия **PV_ROOT** - это специальная привилегия, родительская для всех, кроме **PV_SU_**. Процесс, которому присвоена привилегия **PV_ROOT**, функционирует так, как если бы ему были присвоены все привилегии, кроме **PV_SU_**.

Наборы привилегий процессов:

В ядре определено несколько наборов привилегий, чтобы обеспечить различные виды управления привилегированными операциями. Различные наборы привилегий позволяют операционной системе динамически управлять привилегиями и разрешать приложениям применять принцип наименьших привилегий.

Привилегии связываются с процессом через следующие наборы привилегий:

Ограничительный набор привилегий (LPS)
Определяет жесткое ограничение на привилегии для заданного процесса. Никакая эскалация привилегий в системе не может поднять привилегии процесса выше этого значения. Это означает,

что процесс не может получить больше привилегий, чем это значение, какие бы интерфейсы системы он ни использовал. Иными словами, процесс ограничен этими привилегиями в любой момент времени. Это также означает, что остальные наборы привилегий всегда будут подмножествами LPS. Хотя LPS нельзя расширить, каждый процесс будет иметь право сократить LPS. После сокращения LPS его нельзя будет вернуть к прежнему значению. Понижение LPS позволяет процессу сократить диапазон возможных привилегий. Например, процесс может сократить LPS непосредственно перед запуском пользовательской программы. По умолчанию все имеющиеся в системе привилегии задаются в LPS для процесса.

Максимальный набор привилегий (MPS)
Полный набор привилегий, доступных процессу. MPS может содержать любую привилегию, входящую в LPS, но не может превосходить LPS. За время существования процесса MPS может изменяться многократно по разным причинам. Ниже приведены некоторые из этих причин:

- Когда текущий процесс выполняет другую привилегированную команду и затем получает связанные дополнительные привилегии
- Если у процесса есть необходимые привилегии, он может развернуть MPS программно динамическим образом

Действующий набор привилегий (EPS)
Список текущих активных привилегий процесса. EPS всегда является подмножеством MPS процесса и используется ядром для выполнения проверок доступа применительно к привилегированным операциям. EPS может изменяться процессом и может равняться MPS, но не может превосходить MPS. Процесс может динамически манипулировать EPS для реализации принципа наименьших привилегий. Например, код пользовательского пространства может потенциально повысить бит привилегий контроля в EPS с помощью API **priv_raise**, прежде чем отправить связанный с контролем системный вызов или вызов ядра. Затем привилегию можно понизить с помощью API **priv_lower**, когда связанный с контролем вызов возвратит управление.

Наследуемый набор привилегий (IPS)
Привилегии, передаваемые из родительского процесса в MPS и EPS его дочерних процессов. IPS может содержать любую привилегию, входящую в LPS, но не может превосходить LPS. IPS можно задать в процессе следующими способами:

- Если у процесса есть необходимые привилегии, то он может расширить IPS программно через системный вызов **setppriv**
- При запуске привилегированной команды привилегии, указанные в атрибуте **inheritprivs**, связанном с командой, присваиваются IPS.

Используемый набор привилегий (UPS)
Привилегии, применявшиеся для проверок доступа во время существования процесса. С помощью UPS можно определить привилегии, необходимые процессу. Когда ядро проверяет, есть ли у процесса требуемые привилегии, он сохраняет успешную проверку в UPS для привилегии.

Набор привилегий раздела рабочей нагрузки (WPS)
Системный WPAR не обязательно должен разрешать все те привилегированные операции, которые разрешены в глобальном WPAR. Привилегированные операции, разрешенные в системном WPAR, можно контролировать с помощью WPS. Глобальный root может присвоить WPAR ограниченный набор привилегий с помощью WPS. WPS можно указать в файле конфигурации `/etc/wpar/secattrs` или во время запуска WPAR командой **/usr/sbin/startwpar**. LPS всех процессов, работающих в WPAR, равен их WPS.

С помощью административных команд системный администратор может просматривать и изменять различные наборы привилегий процесса. С помощью команды **lssecattr** можно просмотреть LPS, MPS, EPS, IPS и UPS. С помощью команды **setsecattr** - изменить LPS, MPS, EPS и IPS. Изменить UPS командой **setsecattr** нельзя, поскольку UPS - это атрибут, предназначенный только для чтения.

База данных привилегированных команд:

Права доступа, роли и привилегии позволяют реализовывать детализированные средства защиты. Однако применение RBAC в различных системных операциях обуславливает необходимость использования стратегии защиты RBAC.

Хотя ранее некоторые команды AIX напрямую проверяли права доступа, для выполнения таких проверок требовалось изменить сам исполняемый код. Расширенный режим RBAC предоставляет структуру для реализации проверок прав доступа и предоставления связанных привилегий посредством базы данных привилегированных команд; изменять системные исполняемые файлы при этом не требуется.

База данных привилегированных команд предоставляет пользователям доступ к командам, которые в противном случае они не могли бы выполнить. База данных сохраняет информацию о правах доступа к конкретной команде, а также привилегии, предоставляемые процессу в случае успешной проверки прав доступа. В случае локального размещения базы данных она хранится в файле `/etc/security/privcmds` и содержит информационные разделы в виде атрибутов команда-защита. Ниже приведены некоторые ключевые атрибуты этой базы данных (полное описание всех атрибутов приведено в файле `/etc/security/privcmds`).

accessauths
: Список разрешений на доступ, защищающих выполнение команды. При наличии любых из перечисленных прав доступа пользователь может выполнять команду и некоторые или все привилегированные операции, содержащиеся в этой команде.

innateprivs
: Врожденными называются привилегии, присваиваемые процессу в случае успешной проверки разрешений на доступ у инициатора запуска.

authprivs
: Уполномоченными называются дополнительные привилегии, присваиваемые процессу, если у пользователя есть связанные права доступа. Этот атрибут обеспечивает более детализированное управление командой, позволяя ограниченному набору пользователей выполнять дополнительные привилегированные операции.

inheritprivs
: Наследуемыми называются привилегии, которые процесс передает своим дочерним процессам.

secflags
: Список флагов защиты. FSF_EPS - это флаг, указывающий, что при выполнении команды максимальный набор привилегий (MPS) должен быть загружен в действующий набор привилегий (EPS).

Когда пользователь в расширенном режиме RBAC пытается запустить команду, команда сначала проверяется в базе данных привилегированных команд. Если команда существует в базе данных, то проверяются права доступа, связанные с сеансом пользователя, и значение атрибута **accessauths** команды. Если у сеанса есть какие-либо из перечисленных прав доступа, то пользователь может выполнять команду независимо от того, пройдет ли он проверки выполнения DAC для нее. При вызове у процесса команды будут привилегии, указанные в атрибуте **innateprivs**, присвоенном максимальному набору привилегий (MPS). Дополнительные проверки прав доступа выполняются относительно пар права доступа - привилегии, перечисленных в атрибуте **authprivs**. Если у сеанса есть какие-либо из перечисленных прав доступа, то связанные привилегии также добавляются в MPS процесса команды. Запись команды в базе данных привилегированных команд, для которой задано значение **FSF_EPS** в атрибуте **secflags**, присваивает все привилегии, имеющиеся в MPS, действующему набору привилегий (EPS) в момент вызова команды.

Команда считается привилегированной, когда она находится в базе данных привилегированных команд. Хотя программы setuid, отсутствующие в базе данных, с технической точки зрения являются привилегированными командами, они не называются таковыми при описании функционирования RBAC. Если у команды нет записи в базе данных привилегированных команд, то она не считается

привилегированной и доступ к ней контролируется DAC и самой командой. Кроме того, если команда указана в базе данных привилегированных команд, но у сеанса пользователя нет прав на вызов команды, то система возвращается к проверке доступа DAC и разрешает выполнение команды, только если эти проверки окажутся успешными.

Для работы с базой данных привилегированных команд предусмотрено несколько управляющих команд. Создавать и изменять записи в базе данных привилегированных команд можно командой **setsecattr**, просматривать - командой **lssecattr**, удалять - командой **rmsecattr**.

Определение требуемых прав доступа для команды:

Для надлежащего выполнения многих системных административных приложений требуются права доступа. Хотя в базе данных привилегированных команд указан набор предопределенных команд, системному администратору может понадобиться добавить записи, специфические для среды. База данных привилегированных команд позволяет добавлять записи в базу данных. Правильные права доступа должны быть перечислены в атрибуте **accessauths** в порядке получения доступа к команде.

Существует два способа использования и проверки прав доступа в AIX с помощью усовершенствованной структуры RBAC:

- Access Auths (Access Authorization): Атрибут указывается в базе данных привилегированных команд и содержит список через запятую имени прав доступа. Пользователю, текущий сеанс которого имеет одно из прав доступа в списке, разрешено выполнять команду. Это проверяется загрузчиком системы при выполнении защищенных привилегированных исполняемых программ.
- Check Auths (checkauths()): Определенное право доступа или список прав доступа могут быть проверены программно с помощью API checkauths(). Определенные права доступа проверяются по отношению к правам доступа, которые присутствуют в роли текущего сеанса. На основании результата этой проверки программа может выполнить привилегированные операции.

Перед добавлением команды в базу данных привилегированных команд следует определить набор прав доступа, чтобы обеспечить безопасное выполнение команды. Программа или приложение может выполнять внутри себя дополнительную проверку прав доступа. Необходимо определить список прав доступа, используемый в процессе, который может быть назначен при создании пользовательской роли.

Ниже приведена базовая стратегия определения необходимых для команды прав доступа:

1. Предоставьте вызывающей оболочке права доступа **PV_ROOT** или присвойте роль с правами доступа *aix*.

   Важное замечание: В глобальном WPAR право доступа **PV_ROOT** должно быть назначено для эффективного и максимального набора прав доступа для вызова процесса оболочки. В системном WPAR это право доступа также должно быть добавлено для наследования набора прав доступа процесса.
2. Выполните команду.
3. Запишите права доступа, использованные для процесса.
4. Сохраните права доступа, сообщенные в *Access Auths* в атрибуте **accessauths** команды, в базе данных привилегированных команд. Права доступа, сообщенные в *Check Auths*, можно использовать при создании ролей в системе.

Эти шаги следует выполнять в регулируемой среде, поскольку права доступа **PV_ROOT** предоставляются оболочке, или предполагается роль с правами доступа *aix*, и оба эти метода дают очень широкие полномочия. Кроме того, выполнение команды может оказывать влияние на систему, что затронет других пользователей. На практике это процедура, реализуемая методом проб и ошибок. Для получения полного набора прав доступа может понадобиться многократно запускать команду с различными флагами и опциями, к тому же, возможно, на длительное время для приложений, которые рассчитаны на длительное выполнение. Требуемый набор прав доступа для процесса можно легко получить с помощью одной из следующих процедур, которые могут быть выполнены администратором с необходимыми правами доступа:

traceauth

Принимаемый аргумент - выполняемая команда. Команда **traceauth** запускает команду и записывает оба типа прав доступа, использованных при работе процесса. При завершении своего выполнения команда **traceauth** выводит использованные права доступа на **stdout**.

lssecattr

Если команда является процессом, рассчитанным на длительное время, то для просмотра используемых прав доступа можно использовать команду **lssecattr**. Для того чтобы включить трассировку проверки прав доступа в системе, выполните следующую команду:

setrunmode –c; setsecconf –o traceauth=enableДля того чтобы просмотреть использованные права доступа для процесса, выполните команду **lssecattr** следующим образом, заменив ИД наблюдаемого процесса:

lssecattr –p –A *PID*

После определения набора требуемых прав доступа выполните шаги, описанные в “Добавление команды в базу данных привилегированных команд” на стр. 98, чтобы добавить команду в базу данных привилегированных команд. После этого команду должен запустить пользователь с правами доступа, чтобы проверить, правильно ли она выполняется.

Определение требуемых прав доступа для команды:

Многие приложения требуют определенные права доступа для правильного выполнения. Хотя в базе данных привилегированных команд указан набор предопределенных команд, системному администратору может понадобиться добавить записи, специфические для среды или приложения. База данных привилегированных команд позволяет добавлять записи для команд и связанных с ними прав доступа.

Перед добавлением команды в базу данных привилегированных команд следует определить минимальный набор требуемых прав доступа, чтобы обеспечить как можно более безопасное выполнение команды. Предоставление каких-либо прав доступа помимо необходимых для надлежащего выполнения является нарушением принципа минимальных прав доступа. Поэтому определение минимального набора требуемых прав доступа является важным шагом при в систему привилегированной команды.

Ниже приведена базовая стратегия определения минимума необходимых для команды прав доступа:

1. Information System Security Officer (ISSO) или пользователь с ролью isso может назначать право доступа **PV_ROOT** системному администратору, выполняющему команду, которая должна быть назначена привилегированной базе данных. Назначение права доступа **PV_ROOT** вызывающей оболочке выполняется с помощью команды setsecattr. Например:

   setsecattr -p eprivs=PV_ROOT mprivs=PV_ROOT $$
2. Выполните команду для сбора набора прав доступа.
3. Запишите набор прав доступа, использованных для процесса.
4. Сохраните требуемые права доступа в атрибуте **innateprivs** команды в базе данных привилегированных команд.

Эти шаги следует выполнять в регулируемой среде, поскольку права доступа **PV_ROOT** предоставляются оболочке и при этом дают очень широкие полномочия. Кроме того, выполнение команды может оказывать влияние на систему, что затронет других пользователей. На практике это процедура, реализуемая методом проб и ошибок. Для получения полного набора прав доступа может понадобиться многократно запускать команду с различными флагами и опциями, к тому же, возможно, на длительное время для приложений, которые рассчитаны на длительное выполнение. Требуемый набор прав доступа для процесса можно легко получить с помощью одной из следующих процедур, которые могут быть выполнены администратором с необходимыми правами доступа:

tracepriv

Принимаемый аргумент - выполняемая команда. Команды **tracepriv** запускает команду и

записывает использованные при работе процесса привилегии. При завершении своего выполнения команда **tracepriv** выводит использованные права доступа на **stdout**.

lssecattr

Если команда является процессом, рассчитанным на длительное время, то для просмотра используемых прав доступа можно использовать команду **lssecattr**. Для того чтобы просмотреть список использованных прав доступа, выполните команду следующим образом, заменив ИД наблюдаемого процесса:

lssecattr –p –a uprivs *ИД-процесса*

После определения минимального набора требуемых прав доступа выполните шаги, описанные в "Добавление команды в базу данных привилегированных команд", чтобы добавить команду в базу данных привилегированных команд. После этого команду должен запустить пользователь с правами доступа, чтобы проверить, правильно ли она выполняется.

Эскалация привилегий:

Когда системный вызов **fork** создает новый процесс, **fork** предоставляет процессу те же привилегии, что имеет родительский процесс (процесс, вызвавший **fork**). Когда процесс выполняет системный вызов **exec** для исполняемого файла, **exec** заново вычисляет привилегии для исполняемого файла на основе текущих привилегий **exec** и привилегий исполняемого файла.

Привилегии эскалируются следующим образом:

1. Сначала вычисляется объединение (поразрядное логическое ИЛИ) наследуемых привилегий старого (родительского) процесса и врожденных привилегий исполняемого файла.
2. Если у пользователя есть необходимые права доступа, вычисляется объединение (поразрядное логическое ИЛИ) результата предыдущего шага и уполномоченных привилегий.
3. При наличии ограничительных привилегий вычисляется пересечение результата предыдущего шага и ограничительных привилегий. Ограничительные привилегии, если они есть, наследуются посредством системного вызова **exec**.
4. Набор привилегий, получающийся в результате такого объединения, становится максимальным набором привилегий для нового процесса.
5. Если в исполняемом файле существуют наследуемые привилегии, то они присваиваются набору наследуемых привилегий нового процесса. В противном случае, набор наследуемых привилегий старого (родительского) процесса переносится в набор наследуемых привилегий нового процесса.

Если в исполняемом файле задан флаг защиты файла **FSF_EPS**, то действующий набор привилегий для нового процесса совпадает с максимальным набором привилегий. В противном случае, действующие привилегии нового процесса совпадают с наследуемыми привилегиями старого (родительского) процесса.

Добавление команды в базу данных привилегированных команд:

Перед добавлением команды в базу данных привилегированных команд следует тщательно рассмотреть права доступа, чтобы предоставить ей именно те права, которые необходимы.

Полное описание атрибутов, которые можно использовать с командой, приведено в файле `/etc/security/privcmds`. Чтобы определить, какие атрибуты указать с командой, воспользуйтесь следующими вопросами:

1. Будут ли необходимы права доступа для запуска команды?

   ДА Если права доступа не существуют, создайте их с помощью команды **mkauth**. Укажите права доступа в атрибуте **accessauths**.

   НЕТ Если все пользователи будут иметь право запуска команды, присвойте атрибуту **accessauths** значение **ALLOW_ALL**.

2. Будет ли владельцу группы команды разрешено запускать эту команду даже при отсутствии у него соответствующих прав доступа?

   ДА Укажите в списке прав доступа атрибута **accessauths** значение **ALLOW_OWNER** или **ALLOW_GROUP**.

3. Требует ли команда явного указания набора прав доступа при ее выполнении?

   ДА Чтобы определить требуемые права доступа для атрибута **innateprivs**, запускайте эту команду с различными опциями от лица корневого пользователя с помощью команды **tracepriv**.

4. Будут ли пользователи с особыми правами доступа получать дополнительные привилегии?

   ДА Укажите дополнительные пары прав и привилегий в атрибуте **authprivs**.

5. Должна ли команда вести себя, как программа SUID или SGID?

   ДА Укажите соответственно EUID или EGID.

6. Должны ли права доступа, предоставленные команде, передаваться дочерним процессам?

   ДА Укажите права доступа в атрибуте **inheritprivs**.

7. Будет ли реальный набор прав доступа эквивалентен максимальному набору прав доступа при вызове команды?

   ДА Укажите флаг **FSF_EPS** для атрибута secflags.

   NO Не указывайте атрибут **secflags**. Если флаг **FSF_EPS** не указан, то предполагается, что код команды будет повышать и понижать свои полномочия по мере необходимости.

8. Нужно ли выполнять команду со специальным фактическим ИД пользователя 0?

   ДА Указать атрибут RUID.

9. Является ли команда очень важной и требующей управления и обязательного присутствия более одного человека перед своим вызовом?

   ДА Укажите атрибут **authroles** и назначьте значение со списком ролей. Пользователи каждой роли должны быть аутентифицированы перед выполнением команды.

Ответив на эти вопросы, выполните команду **setsecattr** с атрибутами, необходимыми для добавления команды в базу данных. Если эта команда существует и является командой SUID или SGID, то следует рассмотреть вопрос об удалении битов **SUID** и **SGID** из файла, чтобы применить модель с минимальными правами доступа.

База данных привилегированных устройств:

В базе данных привилегированных устройств хранится список привилегий, которые можно считывать из устройства и записывать в устройство. Эта база данных предоставляет администратору механизм более детального управления доступом к устройству, которым можно управлять обычными средствами.

Если эта база данных хранится локально, она содержится в файле `/etc/security/privdevs`. Привилегии, необходимые для доступа к заданному устройству для чтения или записи, хранятся в следующих атрибутах в базе данных:

readprivs
: Список привилегий, позволяющих считывать из устройства

writeprivs
: Список привилегий, позволяющих записывать в устройство

Когда запрашивается открытие привилегированного устройства в режиме чтения, открытие разрешается только в том случае, если какая-либо из привилегий, указанных в атрибуте **readprivs**, существует в действующем наборе привилегий (EPS) для процесса. Аналогично, если устройство открывается в режиме записи, в EPS должна быть какая-либо из привилегий, указанных в атрибуте **writeprivs**.

Процесс добавления устройства в базу данных привилегированных устройств не является обычной операцией. Для просмотра и обработки базы данных можно воспользоваться командами **lssecattr** и **setsecattr**, но добавление или изменение записей в базе данных требует тщательного изучения. Поскольку права на чтение и запись в устройство контролируются привилегиями, необходимо тщательно изучить команды и приложения, которым необходим доступ к устройству, чтобы убедиться в правильности предоставленных привилегий.

База данных привилегированных файлов:

Многие файлы конфигурации системы в традиционных системах UNIX принадлежат пользователю root, и другие пользователи не могут напрямую изменять их. RBAC позволяет пользователю изменять эти файлы конфигурации системы путем активации роли и запуска команды, позволяющей получить привилегии, необходимые для изменения файла.

У некоторых файлов конфигурации AIX нет интерфейсов команд, позволяющих их изменять. В этих случаях необходим инструмент, позволяющий администратору с соответствующими правами доступа непосредственно редактировать и сохранять такие файлы, недоступные ему иными способами.

База данных привилегированных файлов предоставляет способ определения доступа к файлам конфигурации системы с помощью прав доступа. Если эта база данных хранится локально, то она содержится в файле `/etc/security/privfiles`. Эта база данных отображает файлы конфигурации в права доступа, необходимые для просмотра или изменения этих файлов. Доступ к файлу конфигурации контролируется в этой базе данных следующими атрибутами:

readauths
Список прав на чтение из файла

writeauths
Список прав на запись в файл (права на чтение в этом случае подразумеваются)

Просматривать записи в базе данных привилегированных файлов можно командой **lssecattr**, а создавать и изменять - командой **setsecattr**. Уполномоченные пользователи могут получать доступ к файлам, определенным в базе данных привилегированных файлов, с помощью команды **/usr/bin/pvi**. Команда **pvi** - это привилегированная, служебная версия редактора **vi**, основанная на команде **/usr/bin/tvi**. На команду **pvi** распространяются все те же меры предосторожности, что и на команду **tvi** (например, запрещены флаги **–r** и **-t**, esc-символы оболочки, пользовательские макросы), а также следующие ограничения:

- Система должна находиться в расширенном режиме RBAC.
- Можно открывать только файлы, определенные в базе данных привилегированных файлов.
- В каждый момент времени может быть открыт только один файл.
- Запись в файл, отличный от указанного в командной строке, отключена.
- Файл `/etc/security/privfiles` нельзя редактировать командой **pvi**.
- Открывать ссылки нельзя. Редактировать можно только обычные файлы.

Права доступа проверяются до открытия файла. Если права доступа соответствуют требованиям, то в набор привилегий процесса добавляется **PV_DAC_R** или **PV_DAC_W** (в зависимости от того, открывается файл для чтения или для записи). Если права доступа не соответствуют требованиям, то выдается сообщение об ошибке и пользователь получает отказ в доступе к файлу с помощью команды **pvi**.

Таблицы защиты ядра:

Информация, которая содержится в базах данных прав доступа, ролей, привилегированных команд и привилегированных устройств, не используется для защиты, пока данные не будут загружены в область ядра, определенную как таблицы защиты ядра (KST). В расширенном режиме RBAC проверка прав доступа производится в ядре, поэтому необходимо отправить базы данных ядру перед их использованием.

KST состоит из следующих подтаблиц:

- Таблица прав доступа ядра (KAT)
- Таблица ролей ядра (KRT)
- Таблица команд ядра (KCT)
- Таблица устройств ядра (KDT)

Отправить все или выбранные таблицы из пользовательской области в ядро можно с помощью команды **setkst**. KRT и KCT зависят от KAT, поэтому при выборе обновления KAT таблицы KRT и KCT также будут обновлены, чтобы обеспечить синхронизацию данных в этих таблицах. Для обновления KST предпочтительно создавать или изменять все необходимые базы данных пользовательского уровня (с помощью таких команд как **mkauth**, **chauth**, **mkrole** и **setsecattr**), а затем использовать команду **setkst** для отправки таблиц в ядро. После загрузки таблиц в ядро можно просмотреть содержащуюся в них информацию с помощью команды **lskst**.

Каждая отдельная таблица отправляется в KST целиком. Другими словами, KST не позволяет изменять отдельные записи. Заменять следует всю таблицу. Перед отправкой таблиц в ядро команда **setkst** проверяет таблицы и отношения между ними. Кроме того, команда **setkst** указана в файле `inittab`, чтобы базы данных отправлялись в KST в начале загрузки системы.

Если по каким-либо причинам таблицы не могут быть созданы или загружены в ядро и до этого не была загружена ни одна таблица, система ведет себя таким образом, будто прав доступа и ролей не существует. При отсутствии совпадающих записей команды, API и системные вызовы, связанные с проверкой прав доступа и ролей, возвращают сообщение об ошибке. В этом случае система ведет себя почти так же, как и в обычном режиме RBAC, но что ни один пользователь не получает доступ к коду команд, которые предоставляют права принудительно.

Отключение корневого пользователя:

В режиме расширенного RBAC можно настроить систему таким образом, чтобы с корневым пользователем не были связаны особые полномочия и чтобы система рассматривала его как обычного пользователя.

В прошлом ИД пользователя, равный 0, рассматривался операционной системой как привилегированный ИД, и ему было разрешено обходить принудительные проверки защиты. При отключении корневого пользователя в операционной системе снимаются отметки, которые позволяют пользователю с ИД 0 обходить проверки защиты и требует от процесса прав доступа, с которыми он может пройти проверки защиты. Отключение корневого пользователя снижает ущерб, который может причинить атакующий, до минимума, поскольку в системе больше нет пользователя со всеми правами. После отключения корневого пользователя администрирование системы должно осуществляться пользователями, которым назначены роли с правами доступа.

Права доступа корневого пользователя можно отключить с помощью команды **/usr/sbin/setsecconf**. Для отключения прав доступа корневого пользователя выполните следующую команду, а затем перезапустите систему:

```
setsecconf -o root=disable
```

После выполнения этой команды доступ к учетной записи корневого пользователя с помощью удаленного и локального входа, а также команды su закрыт. Тем не менее, учетная запись корневого пользователя остается владельцем файлов, и при ее открытии будет получен доступ к привилегированным файлам.

В системе с отключенным корневым пользователем процессы, принадлежащие ему, больше не получают никаких особых прав доступа. Об этом нужно помнить, если в системе присутствуют приложения setuid, принадлежащие корневому пользователю, но не внесенные в базу данных привилегированных команд. В этих приложениях setuid, запущенных в среде с отключенным корневым пользователем, вероятно, произойдут сбои, поскольку процесс не сможет выполнять привилегированные операции. В такой системе любую команду, которой необходимо выполнить привилегированные операции, следует добавить в базу

данных привилегированных команд с предоставлением требуемых прав доступа. Поэтому пред отключением корневого пользователя следует тщательно проанализировать систему и приложения, используемые в ней.

Поддержка удаленных баз данных RBAC:

В среде предприятия желательно реализовывать и применять общую стратегию защиты для всех систем среды. Если базы данных, управляющие стратегией, хранятся в различных системах независимо друг от друга, то управление стратегией защиты становится затруднительным для системного администратора. Расширенный режим RBAC AIX позволяет хранить базы данных RBAC на LDAP для централизованного управления всеми системами окружения.

В AIX добавлена поддержка хранения в LDAP всех баз данных, относящихся к RBAC. К RBAC относятся следующие базы данных:

- База данных прав доступа
- База данных ролей
- База данных привилегированных операторов
- База данных привилегированных устройств
- База данных привилегированных файлов

Примечание: База данных прав доступа, хранящаяся в LDAP, содержит только пользовательские права доступа. Права доступа, определенные системой, не могут храниться в LDAP и остаются локальными для каждой клиентской системы.

AIX обеспечивает утилиты для упрощенного экспортирования локальных данных RBAC на LDAP, настройки клиента для использования данных RBAC на LDAP, управления просмотром данных RBAC и управления данными LDAP из клиентской системы. В следующих разделах приведена дополнительная информация о функциях LDAP, которые реализованы в расширенном RBAC.

Экспорт данных RBAC на LDAP:

Подготовку к использованию LDAP в качестве хранилища базы данных RBAC следует начинать с передачи данных RBAC на сервер LDAP.

Для того чтобы клиенты LDAP могли использовать сервер для работы с данными RBAC, на сервере LDAP должна быть установлена схема RBAC для LDAP. Схема RBAC для LDAP расположена в системе AIX а файле `/etc/security/ldap/sec.ldif` . Схему сервера LDAP необходимо обновить с использованием этого файла при помощи команды **ldapmodify**.

Файл **/usr/sbin/rbactoldif** можно использовать для чтения данных из локальных баз данных RBAC и вывода данных в поддерживаемом LDAP формате. Итоговые данные, полученные с помощью команды **rbactoldif** можно сохранить в файл, а затем использовать его для передачи на сервер LDAP с помощью команды **ldapadd**. Для создания данных RBAC для LDAP команда **rbactoldif** использует следующие базы данных:

- `/etc/security/authorizations`
- `/etc/security/privcmds`
- `/etc/security/privdevs`
- `/etc/security/privfiles`
- `/etc/security/roles`

Следует уделить внимание расположению хранилища данных RBAC на сервере LDAP. Рекомендуется располагать данные RBAC на LDAP под тем же родительским DN, что и данные пользователей и групп. ACL данных впоследствии настраиваются таким образом, как этого требует выбранная стратегия защиты.

Конфигурация клиента LDAP для RBAC:

Для использования данных RBAC на LDAP следует настроить систему как клиента LDAP.

Для настройки системы как клиента LDAP можно использовать команду AIX **/usr/sbin/mksecldap**. Команда **mksecldap** производит динамический поиск по указанному серверу LDAP и определяет расположение прав доступа, роли, привилегированной команды, устройства и данных файла, сохраняя результаты в файле `/etc/security/ldap/ldap.cfg`.

После успешной настройки системы как клиента LDAP с помощью команды **mksecldap** необходимо настроить систему для использования LDAP как домен, на котором будет производиться поиск данных RBAC. В файле `/etc/nscontrol.conf` следует добавить в атрибут **secorder** значение LDAP, чтобы задействовать базы данных, которые хранятся на LDAP.

Когда система настроена как клиент LDAP и домен поиска данных RBAC демон клиента **/usr/sbin/secldapclntd** периодически получает данные RBAC от LDAP и отправляет данные в таблицы защиты ядра (KST) с помощью команды **setkst**. С помощью атрибута **rbacinterval** в файле `/etc/security/ldap/ldap.cfg` можно установить для демона период времени получения данных RBAC. По умолчанию этому атрибуту присвоено значение 3600, которое указывает, что получать данные RBAC от LDAP и обновлять KST следует один раз в час. Администратор может обновлять KST и вручную с помощью команды **setkst**.

Управляющий файл службы имен:

Данные RBAC могут находиться только в локальных файлах, только в LDAP или в объединенной базе данных, основанной на локальных файлах и LDAP и настраиваемой в управляющем файле службы имен `/etc/nscontrol.conf`.

Порядок поиска в базах данных прав доступа, ролей, привилегированных команд, устройств и файлов задается по отдельности в файле `/etc/nscontrol.conf`. Порядок поиска в базе данных задается в файле в атрибуте **secorder**, представляющем собой список доменов, разделенных запятыми. Ниже приведен пример конфигурации базы данных прав доступа:

```
authorizations:
        secorder = LDAP,files
```

В этом примере указано, что запросы на права доступа должны сначала просматривать LDAP, а затем, если права доступа не будут найдены в LDAP, - локальные файлы. Доступные системы наборы прав доступа - это объединение прав доступа, предоставляемых LDAP и локальными файлами. Это объединение - не простая комбинация значений из двух доменов, но, скорее, слияние значений. В указанную выше конфигурацию включаются все права доступа из LDAP, а затем добавляются только уникальные права доступа из локальных файлов.

Все изменения и удаления сначала применяются к первому домену списка, и только если сущность не будет найдена в первом домене, они применяются к последующим доменам. В этом случае, сначала проверяется LDAP, а затем, если права доступа не найдены в LDAP, - локальные файлы. Новые сущности всегда создаются в первом из доменов, указанных в атрибуте **secorder**. В приведенном выше примере новые права доступа создаются в базе данных LDAP.

Если для базы данных в файле `/etc/nscontrol.conf` нет записи или файл не существует, то запросы и изменения в базе данных выполняются только в базе данных локальных файлов. Конфигурацию базы данных в файле можно задать командой **chsec** и просмотреть командой **lssec**. Для настройки данных о правах доступа таким образом, чтобы они сначала извлекались из LDAP, а затем - из локальных файлов, выполните следующую команду:

```
chsec -f /etc/nscontrol.conf -s authorizations -a secorder=LDAP,files
```

Конфигурация в файле /etc/nscontrol.conf управляет и интерфейсом библиотеки, и интерфейсом командной строки. Приложения могут извлечь текущее значение атрибута **secorder** для базы данных с помощью интерфейса **getsecorder**. Значение атрибута **secorder** можно переопределить для процесса с помощью интерфейса **setsecorder**.

Включение команд RBAC для LDAP:

Все команды управления базами данных RBAC способны использовать настройки из файла /etc/nscontrol.conf и отправлять запросы, изменять, создавать и удалять субъекты в домене или доменах, определенных для данной базы данных.

По умолчанию домены обрабатываются в соответствии со значением атрибута базы данных **secorder**, но это можно переопределить с помощью опции командной строки **–R**. Когда для команды указана опция **–R**, операция принудительно производится на указанном домене с переопределением настроек, которые содержатся в файле /etc/nscontrol.conf. Для поддержки удаленных доменов используются следующие команды управления базой данных RBAC:

- **mkauth**, **chauth**, **lsauth**, and **rmauth**
- **mkrole**, **chrole**, **lsrole**, and **rmrole**
- **setsecattr**, **lssecattr**, and **rmsecattr**

Для использования настроек из файла /etc/nscontrol.conf используется команда **setkst**. Команда **setkst** получает объединенную копию записей для данной базы данных, как определено в файле, а затем загружает итоговые данные в таблицы защиты ядра.

Междоменное назначение:

При разработке среды, в которой данные RBAC предоставляются двумя доменами, например локальными файлами и сервером LDAP, следует уделить внимание вопросу междоменного назначения элементов. Примерами междоменного назначения служат назначение роли, определенной LDAP, локальному пользователю или назначение локально определенной роли пользователю LDAP.

Назначение удаленного субъекта (роли LDAP) локальному (локальный пользователь) не имеет важного значения, поскольку не влияет на другие системы в среде. В то же время, назначение локального субъекта (локальной роли) удаленному субъекту (пользователю LDAP) следует проводить очень осторожно. Поскольку удаленный субъект (пользователь LDAP) видим на многих клиентах, нет никакой гарантии того, что локальный субъект (локальная роль), назначенная ему, определена или имеет какое-либо определение в каждой клиентской системе. Например, роль может быть определена в каждой системе локально, но с различными правами доступа. Удаленный пользователь, которому назначена эта локальная роль, будет иметь различные права доступа на каждом из этих клиентов, что может иметь нежелательные для защиты последствия.

Во избежание возможных сбоев в защите при назначении субъекту LDAP локального субъекта рекомендуется реализовать на сервере LDAP управление доступом к базам данных RBAC, чтобы ни одна клиентская система не могла изменять записи. Изменение субъектов RBAC на LDAP будет разрешено только тем клиентам, которые соединяются с сервером LDAP с использованием учетной записи с права доступа. Другие клиенты будут иметь только права на чтение из баз данных RBAC на LDAP.

Ограничения на размер в расширенном RBAC:

В следующей таблице перечислены различные ограничения на элементы, связанные с RBAC:

Таблица 10. различные ограничения на элементы, связанные с RBAC

Описание	Максимальный размер
Имя роли	63 печатаемых символа
Максимальное число ролей в сеансе	8
Максимальный размер имени прав доступа	63 печатаемых символа
Максимальное число уровней в иерархии прав доступа	9
Максимальное число разрешений на доступ для команды	8
Максимальное число разрешенных привилегированных наборов для команды	8

Администрирование расширенного RBAC:

В этом разделе описаны общие сценарии использования командной строки для администрирования RBAC. Примеры иллюстрируют основные аспекты функциональности. Также для администрирования RBAC предусмотрены интерфейсы SMIT. Команда быстрого доступа к меню SMIT RBAC - `smit rbac`.

Создание пользовательских прав доступа:

Вы можете создавать пользовательские права доступа, позволяющие управлять выполнением команд.

Создать пользовательские права доступа можно командой mkauth. Изменения в базе данных прав доступа вступают в силу после загрузки изменений в ядро командой **setkst**.

- Для создания пользовательских прав доступа выполните следующую команду:

  mkauth *имя_прав_доступа*

Создание и изменение ролей:

Роль можно создать с помощью команды **mkrole**.

Роли создаются с помощью команды **mkrole**. Изменения в базе данных ролей вступают в силу после их загрузки в ядро с помощью команды **setkst**. Изменять роли можно с помощью команды **chrole**.

- Для создания роли выполните следующую команду:

  mkrole dflt_msg="*Моя роль*" *роль*
- Для создания роли с наследованием прав доступа от существующих ролей выполните следующую команду:

  mkrole rolelist=*дочерняя-роль-1*,*дочерняя-роль-2 роль*
- Для изменения определения роли выполните следующую команду:

  chrole rolelist=*дочерняя-роль-3 роль*

Присвоение прав доступа ролям:

С помощью команд **mkrole** и **chrole** можно присваивать права доступа ролям.

- Следующая команда **mkrole** присваивает права доступа **имя_прав_доступа_1** и **имя_прав_доступа_2** роли **имя_роли**:

  ```
  mkrole authorizations=имя_прав_доступа_1,имя_прав_доступа_2 имя_роли
  ```
- Следующая команда **chrole** присваивает права доступа **имя_прав_доступа_1** и **имя_прав_доступа_2** роли **имя_роли**:

  ```
  chrole authorizations=имя_прав_доступа_1,имя_прав_доступа_2 имя_роли
  ```

Установка режима идентификации для роли:

Активацией ролей можно управлять с помощью атрибута роли **auth_mode**.

Для атрибута **auth_mode** допустимы следующие значения:

NONE Идентификации не требуется

INVOKER
Вызывающий должен ввести свой пароль. Это значение по умолчанию.

Для того чтобы пользователи идентифицировали себя при принятии определенной роли введите следующую команду:

chrole auth_mod=INVOKER *роль*

Назначение ролей для пользователя:

Для назначения пользователям ролей можно использовать команду **chuser**.

Для назначение ролей **роль-1** и **роль-2** пользователю **пользователь** выполните следующую команду:
```
chuser roles=роль-1,роль-2 пользователь
```

Активация ролей:

По умолчанию для выполнения привилегированных команд пользователь должен активировать роль в сеансе.

- Для активации ролей **роль-1** и **роль-2** выполните следующую команду:
  ```
  swrole роль-1,роль-2
  ```
- Некоторые роли, назначаемые пользователям, классифицируются как роли по умолчанию. Эти роли активируются автоматически при входе пользователя в систему. Они остаются активными в течение всего сеанса. Для назначения роли роль-1 ролью по умолчанию для пользователя выполните следующую команду:
  ```
  chuser roles=роль-1,роль-2 default_roles=роль-1 пользователь
  ```

Просмотр набора активных ролей:

С помощью команды **rolelist** с опцией **-e** можно просмотреть информацию о действующем наборе активных ролей для сеанса.

- Для просмотра действующего набора активных ролей для сеанса выполните следующую команду:
  ```
  rolelist -e
  ```

Просмотр ролей пользователя:

Команда **rolelist** выдает информацию о ролях и правах доступа для текущих ролей пользователя или присвоенных им ролей.

По умолчанию команда **rolelist** показывает список ролей, присвоенных пользователю. Это, в общем-то, та же информация, что выдается командой `lsuser -a roles user1`, за исключением того, что она также содержит текстовое описание роли, если оно было указано.

- Для просмотра присвоенных вам ролей и связанных с ними прав доступа выполните следующую команду:
  ```
  rolelist -a
  ```

Контроль ролей сеанса:

Роли, активные в течение сеанса входа, контролируются наряду с такими атрибутами, как UID и GID. Эти роли можно просмотреть с помощью команды **auditpr**.

Для просмотра ролей в контрольном журнале выполните следующую команду:

```
auditpr -h eli -i /audit/trail
```

Предоставление прав доступа выполняющемуся процессу:

Для изменения прав доступа выполняющегося процесса можно использовать команду **setsecattr**.

- Для обновления текущего набора прав доступа, связанного с процессом, выполните следующую команду:

  setsecattr –p eprivs=*права-доступа pid*
- Перед добавлением каких-либо прав доступа к текущему набору процесса следует убедиться, что права доступа уже существуют в максимальном наборе прав доступа. Для изменения максимального набора прав доступа выполните следующую команду:

  setsecattr –p mprivs=*права-доступа pid*

Администрирование прав доступа WPAR:

Каждый WPAR связан с набором прав доступа, определяющим его полномочия. Он называется набором прав доступа WPAR (WPS).

Процессы, выполняющиеся в данном WPAR могут пользоваться только теми правами доступа, которые включены в WPS.

- Для изменения WPS из глобального WPAR выполните следующую команду:

  chwpar –S privs+=*права-доступа имя-wpar*

Определение требуемых прав доступа для команды:

Некоторым командам для выполнения привилегированных операций необходимы права доступа. Права доступа используются ядром для обхода ограничений защиты.

Для того чтобы определить, какие права доступа нужны для успешного выполнения команды и настроить ее соответствующим образом, используйте команду **tracepriv**. Команда **tracepriv** записывает права доступа, используемые другой командой во время выполнения последней. Настраиваемую команду нужно запустить с правами доступа **PV_ROOT**, чтобы все попытки использования прав доступа были завершены успешно. После завершения работы команды набор использованных прав доступа отправляется на `stdout`.

- Для настройки команды выполните следующую команду:

  tracepriv –ef *команда*

Управление командами с помощью прав доступа:

С помощью прав доступа можно управлять выполнением команд.

Связать права доступа с командой можно с помощью команды **setsecattr**. Команда **setsecattr** добавляет раздел в базу данных привилегированных команд (`/etc/security/privcmds`). Изменения в этой базе данных должны загружаться в ядро командой **setkst**.

- Для того чтобы связать права доступа с командой, выполните следующую команду:

  setsecattr –c accessauths=*имена_прав_доступа* **innateprivs=***привилегии* **proxyprivs=***привилегии* **authprivs=***имя_прав_доступа***=***привилегии имя_команды*

Управление доступом к устройствам:

RBAC предоставляет механизм усовершенствованного управления доступа к устройствам. Системный администратор может указать права доступа, требуемые для открытия устройства в режиме чтения или записи.

Например, доступом для записи к записывающему устройству DVD можно управлять с помощью прав доступа **PV_DEV_CONFIG**, и в этом случае права на запись DVD получат только те процессы, для которых указаны эти права.

- Для добавления устройства в базу данных устройств выполните следующую команду:
  setsecattr –d readprivs=*права-доступа* **writeprivs=***права-доступа устройство*

Обновление Таблиц защиты ядра RBAC:

Команда **setkst** считывает базы данных защиты и загружает информацию из баз данных в Таблицы защиты ядра (KST).

По умолчанию все базы данных защиты отправляются в KST. В качестве альтернативы в опции **-t** можно указать конкретную базу данных. Однако, если указано, что в KST должна быть отправлена только база данных прав доступа, то в KST будут также обновлены базы данных ролей и привилегированных команд, поскольку они зависят от базы данных прав доступа.

- Для отправки все последних баз данных RBAC в ядро выполните следующую команду:
  ```
  setkst
  ```

Применение переключателя расширенного режима RBAC:

Переключатель настройки предназначен для отключения расширенного режима RBAC в системе в целом и возврата к обычному поведению RBAC.

Системный администратор может отключить расширенный режим RBAC с помощью команды **chdev** для устройства **sys0**, присвоив атрибуту **enhanced_RBAC** значение **false** и затем перезагрузив систему. Для того чтобы снова переключиться в расширенный режим RBAC следует присвоить атрибуту **enhanced_RBAC** значение **true**, а затем перезагрузить систему.

- Для возврата к обычному режиму RBAC выполните следующую команду:
  ```
  chdev -l sys0 -a enhanced_RBAC=false
  ```
- Для чтения значения атрибута **enhanced_RBAC** выполните следующую команду:
  ```
  lsattr -E -l sys0 -a enhanced_RBAC
  ```

В среде WPAR режим RBAC можно настроить только из глобальной системы. Эти настройки затронут как глобальную систему, так и все WPAR.

Примечание: Отключение расширенного режима RBAC может снизить порог защиты системы, особенно в WPAR.

Команды, связанные с RBAC

В следующей таблице приведены связанные с RBAC команды, существующие в операционной системе AIX для управления средой RBAC и использования ее.

Команда	Описание
chauth	Изменить пользовательские атрибуты прав доступа
chrole	Изменить атрибуты роли
ckauth	Проверить права доступа у текущего процесса

Команда	Описание
lsauth	Показать пользовательские и системные атрибуты прав доступа
lskst	Показать записи Таблиц защиты ядра
lspriv	Показать привилегии, имеющиеся в системе
lsrole	Показать атрибуты роли
lssecattr	Показать атрибуты защиты команды, устройства, процесса или файла
mkauth	Создать пользовательские права доступа
mkrole	Создать роль
pvi	Редактор привилегированных файлов
rbacqry	Включает RBAC для приложений
rbactoldif	Создать вывод баз данных пользовательского уровня RBAC в формате, совместимом с LDAP
rmauth	Удалить пользовательские права доступа
rmrole	Удалить роль
rmsecattr	Удалить определение атрибутов защиты для команды, устройства или файла
rolelist	Показать информацию о роли для пользователя или процесса
setkst	Отправить записи из базы данных пользовательского уровня RBAC в Таблицы защиты ядра
setsecattr	Задать атрибуты защиты команды, устройства, процесса или файла
setsecconf	Изменить флаги защиты ядра
swrole	Создать сеанс роли
tracepriv	Выполнить трассировку привилегий, необходимых для успешного выполнения команды

Файлы, связанные с RBAC

В следующей таблице перечислены связанные с RBAC файлы, предоставляемые в AIX для настройки и сохранения информации базы данных.

Файл	Описание
`/etc/nscontrol.conf`	Управляющий файл службы имен для некоторых баз данных защиты
`/etc/security/authorizations`	База данных пользовательских прав доступа
`/etc/security/privcmds`	База данных привилегированных команд
`/etc/security/privfiles`	База данных привилегированных файлов
`/etc/security/privdevs`	База данных привилегированных устройств
`/etc/security/roles`	База данных ролей

Применение расширенного RBAC в приложениях

Многим приложениям не требуется никаких модификаций для успешной работы в среде расширенного RBAC. Простое определение разрешений на доступ и связанных с ними привилегий и последующее присвоение приложения базе данных привилегированных команд может оказаться достаточным.

Однако приложение может воспользоваться расширенным RBAC, вызвав интерфейсы RBAC для управления выполнением приложения на более детализированном уровне; это делает приложение более защищенным. Ниже указаны приложения, которые могут выиграть от применения расширенного RBAC:

- Приложения, доступные только пользователю root или членам определенной группы. Эти приложения обычно проверяют действующий ИД пользователя или членство в группе; их можно перенастроить на проверку прав доступа.
- Приложения, применяющие биты режима **setuid** или **setgid** с целью разрешить непривилегированным пользователям получать привилегии во время запуска команды. Использование форсирования привилегий обычно делает эти приложения более защищенными, поскольку для выполнения задачи требуется меньше привилегий.

Проверка прав доступа:

Приложения, которые на данный момент для определения полномочий на выполнение привилегированных операций используют ИД пользователя или ИД группы, должны быть изменены таким образом, чтобы вместо этого они проверяли наличие прав доступа.

Рассмотрим, к примеру, приложение, которое выполняет задачи по настройке системы и на данный момент позволяет корневому пользователю (UID = 0) выполнять некоторые привилегированные операции:

```
if (getuid() == 0) {
   /* позволить продолжать выполнение привилегированной операции */
}
```

Для того чтобы приложение вместо этого разрешало выполнять привилегированные операции пользователям с особыми правами доступа (**aix.fs.config**), код можно изменить так, чтобы для проверки прав доступа он использовал API **checkauths**:

```
if (checkauths("aix.fs.config", CHECK_ALL)) {
  /* позволить продолжать выполнение привилегированной операции */
 }
```

API **checkauths** включен как для действующего, так и для расширенного режима RBAC и возвращает код успешного завершения **0**, если вызывающий процесс имеет указанные права доступа. API **checkauths** также определяет, включены ли права доступа корневого пользователя, а затем разрешает или запрещает корневому пользователю обходить проверки прав доступа в соответствии с необходимостью. До AIX версии 6.1 для проверки прав доступа использовались API **MatchAllAuths**, **MatchAnyAuths**, **MatchAllAuthsList** и **MatchAnyAuthsList**. Приложения, представленные в AIX версии 6.1 и выше, должны с этой целью использовать API **checkauths**, поскольку он поддерживает действующий и расширенный режимы RBAC и отключение корневого пользователя.

Как и в приведенном выше примере, в приложениях, которые вызывают **getuid**, **getgid** или подобные функции только для того, чтобы разрешить определенным пользователям выполнять особые задачи, можно использовать API **checkauths** для того чтобы вместо этого проверять права доступа. Если проверяемый ИД пользователя или ИД группы не является ИД корневого пользователя, то сначала с помощью системного вызова **sys_parm** можно узнать, включен ли расширенный RBAC. Если расширенный RBAC не включен, то код может выполнить доступные на данный момент проверки. В противном случае, если расширенный RBAC включен, код может проверить наличие соответствующих системных или пользовательских прав доступа.

Форсирование привилегий:

Помимо возможности проверять права доступа, приложениям теперь можно предоставить и возможность применять детализированное форсирование привилегий во время работы.

Приложения могут повышать привилегии, необходимые для выполнения операции, с помощью API **priv_raise**, и понижать их с помощью API **priv_lower**. Повышение привилегий непосредственно перед выполнением привилегированной операции и понижение привилегий после выполнения операции называется

форсированием привилегий и является предпочтительным способом использования привилегий приложениями. Для повышения привилегии необходимо, чтобы она присутствовала в максимальном наборе привилегий приложения в базе данных привилегированных команд. В случае повышения привилегия попадает в действующий набор привилегий (EPS) процесса. В случае понижения привилегия удаляется из EPS. В следующем примере кода показано форсирование привилегий вокруг API **auditproc**.

```
priv_raise(PV_AU_ADMIN, -1); /* повысить привилегию при необходимости */
auditproc();     /* вызвать системный вызов контроля */
priv_lower(PV_AU_ADMIN, -1); /* понизить привилегию */
```

Приложения с поддержкой RBAC:

Обычно в AIX и в системах со включенным корневым пользователем и расширенным RBAC программы **setuid**, запущенные корневым пользователем или принадлежащие ему (с ИД пользователя=0), которые отсутствуют в базе данных привилегированных команд, всегда получают все права доступа к ядру. Поэтому проверка прав доступа всегда завершается успешно - даже если запрошенные права доступа не находятся в наборе действующих прав доступа процесса (EPS).

Такое поведение до сих пор необходимо для поддержки существующих приложений **setuid**, но может представлять собой угрозу защите, поскольку программа **setuid** получает все права корневого пользователя.

Для обеспечения надлежащего ограничения прав доступа процесса с системе со включенным корневым пользователем и расширенным RBAC в структуру процесса введен новый бит. Если он установлен, то процесс становится процессом, который поддерживает RBAC, и действующий ИД пользователя 0 не предоставляет дополнительных прав доступа. Этот бит можно установить в программе с помощью системного вызова **proc_rbac_op**. Любая программа **setuid**, которая отсутствует в базе данных привилегированных команд, может воспользоваться этой функцией для снижения уязвимости путем сужения прав доступа. Помните, что программы, которые определены в базе данных привилегированных команд, автоматически помечаются как процессы, поддерживающие RBAC, и получают лишь те права, которые перечислены в базе данных.

Следующий код показывает, каким образом приложение может пометить себя как поддерживающее RBAC и затем ограничить права доступа:

```
#include <userpriv.h
#include <sys/priv.h>

privg_t effpriv;

int rbac_flags = SEC_RBACAWARE;

/* Пометить процесс как процесс с поддержкой RBAC. */
proc_rbac_op(-1, PROC_RBAC_SET, &rbac_flags);

/* Установить пустые значения для набора действующих прав доступа. */
priv_clrall(effpriv);
setppriv(-1, &effpriv, NULL, NULL, NULL);

/* Расширить права доступа при необходимости. */
priv_raise(PV_AU_ADMIN, -1);
auditproc();

/* Ограничить права доступа, когда в них уже нет необходимости. */
priv_lower(PV_AU_ADMIN, -1);
```

RBAC API:

API для RBAC, предусмотренные в системе, перечислены в следующей таблице. Более подробная информация приведена в отдельных API.

API	Описание
checkauths	Сравнивает полученный список прав доступа с правами доступа, которые связаны с текущим процессом.
GetUserAuths	Получает набор прав доступа, назначенный текущему процессу.
MatchAllAuths, MatchAllAuthsList, MatchAnyAuths, MatchAnyAuthsList	Сравнивает права доступа. Вместо этих API предпочтительно использовать API checkauths.
getauthattr, putauthattr	Опрашивает или изменяет права доступа, определенные в базе данных прав доступа.
getauthattrs	Получает различные атрибуты прав доступа из базы данных прав доступа.
putauthattrs	Обновляет различные атрибуты прав доступа в базе данных прав доступа.
getcmdattr, putcmdattr	Опрашивает или изменяет информацию о защите команды в базе данных привилегированных команд.
getcmdattrs	Получает различные атрибуты команды из базы данных привилегированных команд.
putcmdattrs	Обновляет различные атрибуты команды в базе данных привилегированных команд.
getdevattr, putdevattr	Опрашивает или изменяет информацию о защите устройства в базе данных привилегированных устройств.
getdevattrs	Получает различные атрибуты устройства из базы данных привилегированных устройств.
putdevattrs	Обновляет различные атрибуты устройства в базе данных привилегированных устройств.
getpfileattr, putpfileattr	Опрашивает или изменяет информацию о защите файла в базе данных привилегированных файлов.
getpfileattrs	Получает различные атрибуты файла из базы данных привилегированных файлов.
putpfileattrs	Обновляет различные атрибуты файла в базе данных привилегированных файлов.
getroleattr, putroleattr	Опрашивает или изменяет роли, определенные в базе данных ролей.
getroleattrs	Получает различные атрибуты роли из базы данных ролей.
putroleattrs	Обновляет различные атрибуты роли в базе данных ролей.
getsecorder	Получает упорядочение доменов для определенных защищенных баз данных.
setsecorder	Задает упорядочение доменов для определенных защищенных баз данных.

Права доступа AIX

Права доступа, используемые в AIX, перечислены в следующей таблице. В ней приведено описание каждого права и соответствующих системных вызовов. Некоторые права составляют иерархию, в которой один набор прав доступа может предоставлять все права, связанные с другим.

При проверке прав доступа система сначала проверяет, предоставлены ли процессу минимальный набор требуемых прав, а затем обрабатывает иерархию, проверяя наличие в ней более широких полномочий. Например, процесс с правом доступа **PV_AU_** автоматически получает права доступа **PV_AU_ADMIN**,

PV_AU_ADD, **PV_AU_PROC**, **PV_AU_READ**, и **PV_AU_WRITE**, а процесс с правом доступа **PV_ROOT** автоматически получает все перечисленные ниже права доступа за исключением **PV_SU_**.

Права доступа	Описание	Ссылка на системный вызов
PV_ROOT	Процессу предоставлены права доступа, эквивалентные набору всех перечисленных ниже прав за исключением PV_SU_ (и подчиненных ему прав доступа)	
PV_AU_ADD	Процессу предоставлены права на запись/добавление контрольной записи	auditlog
PV_AU_ADMIN	Процессу предоставлены права на настройку и опрос системы контроля	audit, auditbin, auditevents, auditobj
PV_AU_PROC	Процессу предоставлены права на получение и установку состояния контроля процесса	auditproc
PV_AU_READ	Процессу предоставлены права на чтение файла, помеченного в Trusted AIX как файл контроля	
PV_AU_WRITE	Процессу предоставлены права на запись и удаление файла, помеченного как файл контроля, и на пометку файла как файла контроля в Trusted AIX	
PV_AU_	Эквивалент комбинации всех указанных выше прав доступа в отношении контроля (PV_AU_*)	
PV_AZ_ADMIN	Процессу предоставлены права на изменение таблиц защиты ядра	sec_setkst
PV_AZ_READ	Процессу предоставлены права на получение таблиц защиты ядра	sec_getkat, sec_getkpct, sec_getkpdt, sec_getkrt, etc.
PV_AZ_ROOT	Заставляет процесс проходить проверки на наличие прав доступа во время выполнения exec() (используется в целях наследования)	
PV_AZ_CHECK	Заставляет процесс проходить все проверки на наличие прав доступа	sec_checkauth
PV_DAC_R	Процессу разрешено переопределять ограничения на чтение DAC	access, creat, accessx, open, read, faccessx, mkdir, getea, rename, statx, _sched_getparam, _sched_getscheduler, statea, listea
PV_DAC_W	Процессу разрешено переопределять ограничения на запись DAC	Многие из указанных выше и setea, write, symlink, _setpri, _sched_setparam, _sched_setscheduler, fsetea, rmdir, removeea
PV_DAC_X	Процессу разрешено переопределять ограничения на выполнение DAC	Многие из указанных выше и execve, symlink, rmdir, chdir, fchdir, ra_execve
PV_DAC_O	Процессу разрешено переопределять ограничения на принадлежность DAC	chmod, utimes, setacl, revoke, mprotect
PV_DAC_UID	Процессу разрешено изменять свой ИД пользователя	setuid, seteuid, setuidx, setreuid, ptrace64

Права доступа	Описание	Ссылка на системный вызов
PV_DAC_GID	Процессу разрешено устанавливать новый или изменять свой ИД группы	setgid, setgidx, setgroups, ptrace64
PV_DAC_RID	Процессу разрешено устанавливать новый или изменять свой ИД роли	setroles, getroles
PV_DAC_	Эквивалент комбинации всех указанных выше прав доступа в отношении DAC (PV_DAC_*)	
PV_FS_MOUNT	Процессу разрешено монтировать и размонтировать файловую систему	vmount, umount
PV_FS_MKNOD	Процессу разрешено создавать файлы любого типа или выполнять системный вызов mknod	mknod
PV_FS_CHOWN	Процессу разрешено изменять принадлежность файла	chown, chownx, fchownx, lchown
PV_FS_QUOTA	Процессу разрешено управлять операциями, связанными с дисковыми квотами	quotactl
PV_FS_LINKDIR	Процессу разрешено создание жесткой ссылки на каталог	link, unlink, remove
PV_FS_CNTL	Процессу разрешено осуществлять различные управляющие операции с файловой системой за исключением расширения и сжатия	fscntl
PV_FS_RESIZE	Процессу разрешено осуществлять операции расширения и сжатия файловой системы	fscntl
PV_FS_CHROOT	Процессу разрешено изменять свой корневой каталог	chroot
PV_FS_PDMODE	Процессу разрешено создавать и задавать разделенные каталоги	pdmkdir
PV_FS_	Эквивалент комбинации всех указанных выше прав доступа в отношении файловой системы (PV_FS_*)	
PV_PROC_PRIV	Процессу разрешено изменять и просматривать наборы прав доступа, связанные с процессом	setppriv, getppriv
PV_PROC_PRIO	Процессу/нити разрешено изменять приоритет, стратегию и другие параметры расписания	_prio_requeue, _setpri, _setpriority, _getpri, _sched_setparam, _sched_setscheduler, _thread_setsched, thread_boostceiling, thread_setmystate, thread_setstate
PV_PROC_CORE	Процессу разрешено создавать дамп памяти	gencore
PV_PROC_RAC	Процессу разрешено создавать больше процессов, чем указано в ограничении на одного пользователя	appsetrlimit, setrlimit64, mlock, mlockall, munlock, munlockall, plock, upfget, upfput, restart, brk, sbrk

Права доступа	Описание	Ссылка на системный вызов
PV_PROC_RSET	Разрешено прикреплять набор ресурсов (rset) к процессу или нити	bindprocessor, ra_attachrset, ra_detachrset, rs_registername, rs_setnameattr, rs_discardname, rs_setpartition, rs_getassociativity, kra_mmapv
PV_PROC_ENV	Процессу разрешено задавать информацию о пользователе в структуре пользователя	ue_proc_register, ue_proc_unregister, usrinfo
PV_PROC_CKPT	Процессу разрешено проверять или перезапускать другой процесс	setcrid, restart
PV_PROC_CRED	Процессу разрешено задавать атрибуты разрешений	__pag_setvalue, __pag_setvalue64, __pag_genpagvalue
PV_PROC_SIG	Процессу разрешено посылать сигналы процессу, который к нему не относится	_sigqueue, kill, signohup, gencore, thread_post, thread_post_many
PV_PROC_TIMER	Процессу разрешено предоставлять и использовать таймеры высокой дискретности	appresabs, appresinc, absinterval, incinterval, _poll, _select _timer_settime
PV_PROC_RTCLK	Процессу разрешен доступ к часам CPU	_clock_getres, _clock_gettime, _clock_settime, _clock_getcpuclockid
PV_PROC_VARS	Процессу разрешено получать и обновлять настраиваемые параметры процесса	smttune
PV_PROC_PDMODE	Процессу разрешено изменять режим REAL разделенного каталога	setppdmode
PV_PROC_	Эквивалент комбинации всех указанных выше прав доступа в отношении процессов (PV_PROC_*)	
PV_TCB	Процессу разрешено изменять путь к защищенной библиотеке ядра	chpriv, fchpriv
PV_TP	Указывает, что процесс является процессом защищенного пути и разрешает совершать действия, позволенные процессам защищенного пути. (примечание: те же, что и прежние права доступа AIX BYPASS_TPATH)	
PV_WPAR_CKPT	Процессу разрешено выполнять операцию проверки/перезапуска в WPAR	smcr_proc_info, smcr_exec_info, smcr_mapinfo, smcr_net_oper, smcr_procattr, aio_suspend_io, aio_resume_io
PV_KER_ACCT	Процессу разрешено выполнять операции, доступ к которым ограничен, и которые относятся к подсистеме учета	acct, _acctctl, projctl
PV_KER_DR	Процессу разрешено вызывать операции динамического изменения конфигурации	_dr_register, _dr_notify, _dr_unregister, dr_reconfig
PV_KER_TIME	Процессу разрешено изменять системные часы и системное время	adjtime, appsettimer, _clock_settime

Права доступа	Описание	Ссылка на системный вызов
PV_KER_RAC	Процессу разрешено использовать большие (без страничной организации) страницы для сегментов общей памяти	shmctl, vmgetinfo
PV_KER_WLM	Процессу разрешено инициализировать и изменять конфигурацию WLM	_wlm_set, _wlm_tune, _wlm_assign
PV_KER_EWLM	Процессу разрешено инициализировать и опрашивать среду eWLM	
PV_KER_VARS	Процессу разрешено читать и задавать настраиваемые параметры ядра для времени выполнения	sys_parm, getkerninfo, __pag_setname, sysconfig, kunload64
PV_KER_REBOOT	Процессу разрешено завершать работу системы	reboot
PV_KER_RAS	Для процесса разрешены настройка и запись записей RAS, протокола ошибок, трассировка, создание дампов	mtrace_set, mtrace_ctl
PV_KER_LVM	Процессу разрешено настраивать подсистему LVM	
PV_KER_NFS	Процессу разрешено настраивать подсистему NFS	
PV_KER_VMM	Процессу разрешено изменять параметры обмена и другие настраиваемые параметры VMM в ядре	swapoff, _swapon_ext, vmgetinfo
PV_KER_WPAR	Процессу разрешено настраивать раздел рабочей схемы	brand, corral_config, corral_delete, corral_modify, wpar_mkdevexport, wpar_rmdevexport, wpar_lsdevexport
PV_KER_CONF	Процессу разрешено выполнять различные операции по настройке системы	sethostname, sethostid, unameu, setdomainname
PV_KER_EXTCONF	Процессу разрешено выполнять различные задачи по настройке в расширениях ядра (для служб расширений ядра)	
PV_KER_IPC	Процессу разрешено повышать значение для буфера очереди сообщений IPC и разрешать shmget с присоединением сегментов	msgctl, shm_open, shmget, ra_shmget, ra_shmgetv, shmctl
PV_KER_IPC_R	Процессу разрешено чтение очереди сообщений IPC, массива семафоров и сегментов общей памяти	msgctl, __msgrcv, _mq_open, semctl, shmat, shm_open, __semop, shmctl, __semtimedop, sem_post, _sem_wait, __msgrcv, __msgxrcv
PV_KER_IPC_W	Процессу разрешена запись в очередь сообщений IPC, массив семафоров и сегменты общей памяти	_mq_open, shmat, _sem_open, semctl, shm_open, shmctl, mq_unlink, sem_unlink, shm_unlink, msgctl, __msgsnd
PV_KER_IPC_O	Процессу разрешено переопределять принадлежность DAC для всех объектов IPC	msgctl, semctl, shmctl, fchmod, fchown

Права доступа	Описание	Ссылка на системный вызов
PV_KER_SECCONFIG	Процессу разрешено устанавливать флаги защиты ядра	sec_setsecconf, sec_setrunmode, sec_setsyslab, sec_getsyslab
PV_KER_PATCH	Процессу разрешено исправлять расширения ядра	
PV_KER_	Эквивалент комбинации всех указанных выше прав доступа в отношении ядра (PV_KER_*)	
PV_DEV_CONFIG	Процессу разрешено настраивать расширения ядра и устройства системы	sysconfig
PV_DEV_LOAD	Процессу разрешено загружать и выгружать расширения ядра и устройства системы	sysconfig
PV_DEV_QUERY	Процессу разрешено опрашивать модули ядра	sysconfig
PV_SU_ROOT	Процессу предоставлены все права, связанные со стандартным администратором AIX	
PV_SU_EMUL	Процессу предоставлены все права, связанные со стандартным администратором AIX при ИД пользователя, равном 0	
PV_SU_UID	При этом значении системный вызов getuid возвращает значение 0	getuidx
PV_SU_	Эквивалент комбинации всех указанных выше прав, относящихся к администратору (PV_SU_*)	
PV_NET_CNTL	Процессу разрешено изменять таблицы сетей	socket, bind, listen, _naccept, econnect, ioctl, rmsock, setsockopt
PV_NET_PORT	Процессу разрешено связываться с привилегированными портами	bind
PV_NET_RAWSOCK	Процессу разрешен непосредственный доступ к сетевому уровню	socket, _send, _sendto, sendmsg, _nsendmsg
PV_NET_CONFIG	Процессу разрешено настраивать параметры доступа к сети	
PV_NET_	Эквивалент комбинации всех указанных выше прав доступа в отношении доступа к сети (PV_NET_*)	

Права, указанные в следующей таблице, являются специфическими для Trusted AIX:

Права доступа Trusted AIX	Описание	Ссылка на системный вызов
PV_LAB_CL	Процессу разрешено изменять SCL субъекта в соответствии с допуском процесса	
PV_LAB_CLTL	Процессу разрешено изменять TCL процесса в соответствии с допуском процесса	
PV_LAB_LEF	Процессу разрешено чтение файла кодирования меток	

Права доступа Trusted AIX	Описание	Ссылка на системный вызов
PV_LAB_SLDG	Процессу разрешено понижать уровень SCL процесса в соответствии с допуском процесса	
PV_LAB_SLDG_STR	Процессу разрешено понижать уровень SL пакета в соответствии с допуском процесса	
PV_LAB_SL_FILE	Процессу разрешено изменять SL объекта в соответствии с допуском процесса	
PV_LAB_SL_PROC	Процессу разрешено изменять SL субъекта в соответствии с допуском процесса	
PV_LAB_SL_SELF	Процессу разрешено изменять собственный SL в соответствии с допуском процесса	
PV_LAB_SLUG	Процессу разрешено повышать уровень SL в соответствии с допуском процесса	
PV_LAB_SLUG_STR	Процессу разрешено повышать уровень SL пакета в соответствии с допуском процесса	
PV_LAB_TL	Процессу разрешено изменять TL субъекта объекта	
PV_LAB_	Эквивалент комбинации всех указанных выше прав доступа в отношении меток (PV_LAB_*)	
PV_MAC_CL	Процессу разрешено обходить ограничения допуска секретности	
PV_MAC_R_PROC	Процессу разрешено обходить ограничения на чтение MAC при получении информации о процессе при условии, что метка целевого процесса находится в пределах допуска действующего процесса	
PV_MAC_W_PROC	Процессу разрешено обходить ограничения на запись MAC при отправке сигналов процессу при условии, что метка целевого процесса находится в пределах допуска действующего процесса	
PV_MAC_R	Процессу разрешено обходить ограничения на чтение MAC	
PV_MAC_R_CL	Процессу разрешено обходить ограничения на чтение MAC, когда метка объекта находится в пределах допуска процесса	
PV_MAC_R_STR	Процессу разрешено обходить ограничения на чтение MAC при чтении сообщения из STREAM при условии, что метка целевого процесса находится в пределах допуска процесса	

Права доступа Trusted AIX	Описание	Ссылка на системный вызов
PV_MAC_W	Процессу разрешено обходить ограничения на запись MAC	
PV_MAC_W_CL	Процессу разрешено обходить ограничения на запись MAC, когда метка объекта находится в пределах допуска процесса	
PV_MAC_W_DN	Процессу разрешено обходить ограничения на запись MAC, когда метка процесса поглощает метку объекта и метка объекта находится в пределах допуска процесса	
PV_MAC_W_UP	Процессу разрешено обходить ограничения на запись MAC, когда метка процесса поглощена меткой объекта и метка объекта находится в пределах допуска процесса	
PV_MAC_OVRRD	Позволяет обходить ограничения MAC в отношении файлов, отмеченных как файлы, не включенные в MAC	
PV_MAC_	Эквивалент комбинации всех указанных выше прав доступа в отношении MAC (PV_MAC_*)	
PV_MIC	Процессу разрешено обходить ограничения целостности	
PV_MIC_CL	Процессу разрешено обходить ограничения допуска целостности	

Доменное ролевое управление доступом

Ролевое управление доступом (RBAC), введенное в AIX 6.1, предоставляет механизм разбиения различных функций главного пользователя root на роли, которые могут быть предоставлены другим пользователям в системе. RBAC позволяет распределить обязанности и повышает безопасность системы, поскольку упрощает контроль и отслеживание операций в системе. RBAC позволяет передать ответственность другому пользователю (называемому уполномоченным), но не предоставляет механизма ограничения административных прав уполномоченного пользователя на конкретные ресурсы системы. Например, пользователь с сетевыми административными правами может управлять любым сетевым интерфейсом в системе. Запретить такому пользователю изменять набор интерфейсов нельзя.

Доменная функция RBAC позволяет ограничить доступ уполномоченных пользователей. Пользователи и ресурсы системы помечаются специальными тегами - доменами, и конкретные правила доступа определяют доступ пользователей к ресурсам.

Определения

С правилами доступа связаны следующие определения:

субъект: субъект - это сущность, запрашивающая доступ к объекту. Примером субъекта может служить процесс.

объект: объект - это сущность, содержащая ценную информацию. Примерами объектов могут служить файлы, устройства и сетевые порты.

домен: домен определяется как категория, в которую входит сущность. Если сущность входит в домен, то управление доступом к ней контролируется следующими правилами доступа:

Правила доступа

- Субъект может обращаться к объекту, если субъекту принадлежат все домены, в которые входит объект. Это означает, что список доменов, в которые входит субъект, содержит в себе список доменов объекта как подмножество. Это поведение по умолчанию.
- Субъект может обращаться к объекту, если ему принадлежит хотя бы один домен из числа тех, в которые входит объект. Это означает, что у субъекта и объекта есть хотя бы один общий домен. Это поведение зависит от флагов защиты объекта.
- Объект может запрещать доступ к определенным доменам. Если объект определяет набор доменов, называемый конфликтным набором, и если один из доменов субъекта входит в конфликтный набор, то объект может запрещать доступ к субъекту.

База данных доменов

Домен, поддерживаемый системой, должен храниться в файле конфигурации в каталоге `/etc/security/domains`. Этот раздел задается в следующем формате:

```
имя-домена:
id = <номер>
dfltmsg = <сообщение>
msgcat = <каталог сообщений>
msgset = <набор сообщений в каталоге>
msgnum = <ИД сообщения в каталоге>
```

Базой данных можно управлять с помощью команд **mkdom** и **chdom**. Для просмотра базы данных воспользуйтесь командой **lsdom**. Для удаления записей воспользуйтесь командой **rmdom**.

Записи в базе данных вступают в силу лишь после ее загрузки в ядро, выполняемой с помощью команды **setkst**.

В системе может существовать не более 1024 доменов, и максимально возможный идентификатор домена (атрибут `ID`) равен 1024.

Объекты, назначенные доменам

Для того чтобы назначить домен объекту, необходимо, чтобы объект был определен в базе данных объектов, назначенных доменам. Домены для всех сущностей в системе хранятся в файле конфигурации в каталоге `/etc/security/domobjs`. Этот раздел задается в следующем формате; данный фрагмент служит примером назначения домена объекту:

```
/dev/hrvg:
domains=HR,IT
conflictsets=payroll
objtype=device
secflags=FSF_DOM_ANY
```

domains: задает домены, которым разрешен доступ к объекту. Примерами доменов могут служить IT, HR и Payroll.

objtype: указывает тип объекта, назначаемого домену. Примерами типов объектов могут служить device, file, netint и netport.

conflictsets: указывает, что если субъект принадлежит какому-либо из доменов, заданных в этом атрибуте в этом наборе, то ему не разрешен доступ к объекту.

secflags: этот флаг задает специальные свойства объекта. Можно указать **FSF_DOM_ANY** или **FSF_DOM_ALL**. Если задан флаг **FSF_DOM_ANY**, то субъект может обращаться к объекту, если он содержит какой-либо из доменов, указанных в перечне в атрибуте domains. Если задан флаг **FSF_DOM_ALL**, то для доступа к объекту необходимо, чтобы субъект содержал все домены из этого перечня. Значение по умолчанию - **FSF_DOM_ALL**. Атрибут **secflag** влияет только на поведение атрибута domains объекта.

Домены можно назначать файлам в файловых системах. По умолчанию все домены объекта должны быть подмножеством доменов процесса, чтобы процесс мог обращаться к объекту.

1. Устройства: домену можно назначить любые устройства (включая файловые системы). Проверки домена выполняются во время управляющих операций, например настройки устройства.

   ```
   /dev/hrvg:
   domains=HR,IT
   conflictsets=payroll
   objtype=device
   secflags=FSF_DOM_ANY
   ```

2. Сетевые интерфейсы: когда домену назначаются сетевые интерфейсы (например, en0), управляющие операции, например закрытие интерфейса, потребуют, чтобы интерфейс прошел проверку домена.

   ```
   en0:
   domains=NETIF,ADMIN
   objtype=netint
   flags=FSF_DOM_ALL
   ```

3. Сетевые порты: домену можно назначить порты TCP и UDP. При попытке приложения связаться с портом выполняется проверка домена.

   ```
   TCP_<порт>:
   domains=NETIF,ADMIN
   type=netport
   flags=FSF_DOM_ALL
   ```

4. Процессы: процесс наследует домены пользователя, от имени которого он выполняется. Когда пользователь входит в систему, процесс оболочки пользователя становится владельцем его доменов. Если домены процесса заданы, то они будут действительны в течение всего своего срока существования. Домены процесса нельзя изменить ни с помощью пользовательского интерфейса, ни с помощью системного вызова. Единственный процесс, позволяющий задавать домены, - это процесс входа в систему. У процессов нет атрибутов **conflict set** и **secflags**.

Текущие ограничения

Ниже перечислены ограничения текущих функций домена RBAC:

- В настоящее время файлы конфигурации домена поддерживаются в локальной системе, но не на сервере Упрощенного протокола доступа к каталогам (LDAP).
- Домены RBAC не поддерживаются в разделах рабочей схемы AIX (WPAR).
- Домены RBAC нельзя применить к временным файлам.

Требования к расширенному RBAC

Доменное ролевое управление доступом (RBAC) создано на основе расширенного RBAC; для работы с ним необходимо включить расширенное RBAC в системе.

Таблицы защиты ядра

Домены и объекты, назначенные доменам, которые определены в базе данных доменов и базе данных объектов доменов, вступают в силу после загрузки в ядро, выполняемой с помощью команды **setkst**. Соответствующие две таблицы называются Таблицей доменов ядра (KDOMT) и Таблицей объектов доменов ядра (KDOT).

Дополнительные сведения о таблицах защиты ядра и **setkst** приведены в разделе Ролевое управление доступом (RBAC) руководства по защите AIX.

Команды домена

В следующей таблице перечислены связанные с RBAC команды домена, предоставляемые в операционной системе AIX для управления и использования структуры RBAC домена:

Команда	Описание
mkdom	Создает новый домен
lsdom	Показывает атрибуты домена
rmdom	Удаляет домен
chdom	Изменяет атрибуты домена
setsecattr	Задает атрибуты защиты базы данных объектов доменов
lssecattr	Показывает атрибуты защиты базы данных объектов доменов
rmsecattr	Удаляет определение базы данных объектов доменов
setkst	Отправляет записи из базы данных пользовательского уровня RBAC домена в Таблицы защиты ядра

Связанные с RBAC файлы домена

В следующей таблице перечислены связанные с RBAC файлы, предоставляемые в операционной системе AIX для настройки и хранения информации базы данных:

Файл	Описание
`/etc/security/domains`	База данных доменов
`/etc/security/domobjs`	База данных объектов доменов

Работа с доменами

Определение доменов: домены определяются в базе данных доменов с помощью команды **mkdom**.

```
mkdom id=24 HR
```

Назначение доменов: домены можно назначать сущностям, таким как пользователи, файлы, устройства, сетевые порты и интерфейсы. Все сущности, отличные от пользователей, поддерживают конфликтные наборы и флаги защиты (**secflags**).

Пользователи: пользователи назначаются доменам с помощью команд **chuser** и **chsec**.

Формат:

```
chuser domains = <список
доменов через запятую> имя_пользователя
```

Пример:

```
chuser domains=INET john
```

Во время входа в систему домены, назначенные пользователю, активируются. Если домены изменились в то время, пока ваш сеанс был активен, то необходимо заново войти в систему, чтобы новые домены вступили в силу.

Объекты: для того чтобы запретить доступ к объектам через домены, необходимо, чтобы объект был определен в базе данных объектов доменов; для этого служит команда **setsecattr**.

Формат:

```
setsecattr -o domains=<список разрешенных
доменов через запятую>
conflictsets=<список запрещенных доменов через
запятую>
secflags=<FSF_DOM_ALL или FSF_DOM_ANY>
objtype=<file, или device, или netint, или
netport>
путь_к_объекту
```

Пример:

```
setsecattr -o domains=INET,WEB conflictsets=DB
secflags=FSF_DOM_ANY objtype=netint en0
```

Списки управления доступом

Как правило, ACL представляет собой набор записей, называемых Записями управления доступом (ACE). Каждая запись ACE определяет права доступа пользователя к отдельному объекту.

Каждый раз при обращении к объекту операционная система в соответствии ACL, связанным с этим объектом, проверяет, обладает ли пользователь достаточными правами доступа. ACL в сочетании со связанными проверками прав доступа - это основа механизма Самостоятельного контроля доступа (DAC), поддерживаемого операционной системой AIX.

В операционной системе поддерживается несколько типов системных объектов, с помощью которых процессы могут хранить и передавать информацию. Ниже перечислены основные типы объектов, доступ к которым предоставляется на основе прав доступа:

- Файлы и каталоги
- Конвейеры с именами
- Объекты IPC, такие как очереди сообщений, сегменты общей памяти и семафоры

Все проверки прав доступа к объектам выполняются на уровне системных вызовов при первом обращении к объекту. Поскольку обращение к объектам System V Interprocess Communication (SVIPC) выполняется без учета состояний этих объектов, то проверка выполняется при каждом обращении. Для объектов, имена которых соответствуют именам файловых систем, необходимо также обеспечить преобразование имени в фактическое расположение объекта. Имена преобразуются либо относительно (по отношению к рабочему каталогу процесса), либо абсолютно (по отношению к корневому каталогу процесса). Все операции преобразования имен начинаются с определения одной из этих точек отсчета.

Дискреционный механизм управления доступом обеспечивает эффективное управление доступом к информационным ресурсам и позволяет разделить защиту конфиденциальности и целостности информации. Эффективность механизма управления доступом от имени владельца определяется исключительно тем, как пользователи используют этот механизм. Все пользователи должны понимать принципы настройки, проверки и предоставления прав доступа.

Например, ACL, связанный с объектом файловой системы (файлом или каталогом), позволяет применить права доступа для разных пользователей. Кроме того, ACL можно настроить таким образом, чтобы разным пользователям предоставлялись различные права доступа, такие как чтение и запись.

Как правило, для каждого объекта в качестве владельца указывается конкретный пользователь или основная группа. Владелец задает значения атрибутов доступа к объекту. Атрибуту владельца присваивается действующий ИД пользователя процесса, создавшего объект.

В следующем списке перечислены атрибуты управления прямым доступом для различных типов объектов:

Владелец

Для объектов System V Interprocess Communication (SVIPC) изменять владельца может либо создатель, либо текущий владелец. С каждым объектом SVIPC связан создатель, у которого есть все права доступа владельца (включая права на предоставление доступа). Создателя нельзя изменить даже при наличии прав доступа root.

Объектам SVIPC присваивается действующий ИД группы процесса, создавшего объект. Для объектов файловых систем атрибуты управления прямым доступом инициализируются в соответствии с действующим ИД группы или ИД группы родительского каталога (в зависимости от флага наследования группы родительского каталога).

Группа

Группу может изменить владелец объекта. Новая группа должна быть либо действующей группой процесса-создателя, либо группой родительского каталога. (Объектам SVIPC соответствует группа создателя, которую нельзя изменить. Права доступа этой группы совпадают с правами доступа группы объекта.)

Режим Команда **chmod** (в цифровом формате с восьмеричной записью) позволяет задавать базовые права доступа и атрибуты. Функция **chmod**, вызываемая этой командой, отключает расширенные права доступа. При вызове команды **chmod** с числовым аргументом для файла с ACL расширенные права доступа к этому файлу отключаются. Команда **chmod** с атрибутом, заданным в символьном формате, отключает расширенные права доступа ACL для типа NFS4, но не отключает расширенные права доступа ACL для типа AIXC. Информация о записи режима доступа в числовом и символьном форматах приведена в описании команды **chmod**.

Многие объекты операционной системы, такие как сокеты и объекты файловой системы, обладают ACL, связанными с разными субъектами. Сведения об ACL объектов таких типов могут отличаться друг от друга.

Для управления доступом к объектам файловой системы в AIX применяются биты режима. Кроме того, поддерживались уникальные ACL, связанные с разрядами режима. Такие ACL состояли из базовых битов режима и позволяли определять несколько записей ACE; каждая запись ACE определяла права доступа пользователя и группы для битов режима. Поддержка этого классического поведения ACL продолжится и в следующих версиях. Он называется AIXC ACL.

Обратите внимание, что поддержка ACL для объектов файловой системы зависит от базовой физической файловой системы (PFS). PFS должна поддерживать данные ACL, а также иметь возможность загружать, сохранять и применять информацию о правах доступа различных пользователей. В некоторых файловых системах поддержка ACL не реализована совсем (либо поддержка только основных битов режима) по сравнению с файловыми системами, поддерживающими ACL нескольких типов. В нескольких файловых системах, применяемых в AIX, реализована поддержка ACL нескольких типов. JFS2 и GPFS поддерживают ACL протокола NFS версии 4. В AIX такие списки управления доступом называются ACL типа NFS4. ACL этого типа практически полностью соответствуют определению ACL в спецификациях протокола NFS версии 4. Кроме того, ACL типа NFS4 поддерживает более дискретные средства управления доступом по сравнению с ACL типа AIXC и обеспечивает такие возможности, как наследование.

Поддержка структуры нескольких типов списков управления доступом

Начиная с версии 5.3.0, операционная система AIX поддерживает инфраструктуру списков управления доступом (ACL), позволяющую применять ACL различных типов для разных объектов файловой системы.

Такой подход позволяет управлять ACL разных типов с помощью единого набора методов. В состав этой среды входят следующие компоненты:

Команды администрирования ACL

Это такие команды, как **aclget**, **aclput**, **acledit**, **aclconvert** и **aclgetttypes**. Эти команды обращаются к библиотечным интерфейсам, которые вызывают предназначенные для данного типа ACL модули.

Библиотечные интерфейсы ACL

Библиотечные интерфейсы ACL работают как внешние интерфейсы прикладных программ, которым необходим доступ к спискам ACL.

Определенные для конкретного типа ACL динамически загружаемые модули

В операционной системе AIX представлен ряд модулей, предназначенных для классических списков управления доступом AIX (**AIXC**) и списком управления доступом типа NFS4 (**nfs4**).

Двоичная совместимость:

Как правило, с приложениями, работающими в существующих файловых системах JFS2 со списками ACL AIX или без таких списков, проблем совместимости не возникает.

Однако обратите внимание, что при обращении к объектам файловой системы, с которыми связаны строгие ACL (такие как NFS4), приложениям может быть отказано в доступе. Например, даже для простой проверки существования файла, с которым связан ACL типа NFS4, необходимы права доступа на чтение.

Типы списков управления доступом, поддерживаемые в операционной системе AIX

В настоящее время AIX поддерживает ACL типа AIXC и NFS4.

Кроме того, в AIX реализована инфраструктура, позволяющая применять ACL других типов, поддерживаемых основной физической файловой системой. Обратите внимание, что JFS2 PFS поддерживает ACL типа NFS4 по умолчанию, если конкретный экземпляр файловой системы создан с помощью функции Расширенные атрибуты версии 2.

Списки управления доступом типа AIXC:

Списки управления доступом типа AIXC работают так же, как работали ACL в выпусках AIX до 5.3.0. Списки ACL типа AIXC включают базовые и расширенные права доступа.

Тип списка управления доступом (ACL) AIXC работает так же, как работали ACL в выпусках AIX до 5.3.0. Списки ACL типа AIXC включают базовые и расширенные права доступа. В файловой системе JFS2 максимальный размер списков ACL типа AIXC составляет 4 Кб.

Настройка базовых прав доступа для ACL типа AIXC

Базовые права доступа - это традиционные режимы доступа к файлу, указанные для владельца файла, для группы файла и для остальных пользователей. Поддерживаются следующие режимы доступа: чтение (r - read), запись (w - write) и выполнение/поиск (x - execute).

В списках управления доступом базовые права доступа указываются параметром *режим* в виде rwx (дефис (-) означает отсутствие соответствующих прав доступа):

```
base permissions:
   owner(name): режим
   group(group): режим
   others: режим
```

Настройка атрибутов для ACL типа AIXC

В списки управления доступом типа AIXC можно добавить следующие атрибуты:

setuid (SUID)
: Бит режима Set-user-ID. Этот атрибут задает в качестве действующего и сохраненного ИД пользователя процесса ИД владельца выполняемого файла.

setgid (SGID)
: Бит режима Set-group-ID. Этот атрибут задает в качестве действующего и сохраненного ИД пользователя процесса ИД группы выполняемого файла.

savetext (SVTX)
: Для каталогов указывает, что создавать и удалять связи с файлами этого каталога могут только владельцы этих файлов.

Эти атрибуты добавляются в следующем формате:

```
attributes: SUID, SGID, SVTX
```

Настройка расширенных прав доступа для ACL типа AIXC

Расширенные права доступа позволяют владельцу файла более точно определять доступ к файлу. Расширенные права доступа изменяют базовые права доступа к файлу (владелец, группа и прочие

пользователи), разрешая, запрещая или изменяя режимы доступа для отдельных пользователей, групп и сочетаний групп. Для изменения прав доступа применяются ключевые слова.

Ключевые слова **permit**, **deny** и **specify** определяются следующим образом:

permit Предоставляет пользователю или группе указанный способ доступа к файлу

deny Запрещает пользователю или группе указанный способ доступа к файлу

specify Точно определяет права доступа пользователя или группы к файлу

Если с помощью ключевого слова **deny** или **specify** пользователю запрещен какой-либо способ доступа к файлу, то переопределить этот запрет с помощью других ключевых слов нельзя.

Для применения расширенных прав доступа в ACL должно быть указано ключевое слово **enabled**. По умолчанию указывается ключевое слово **disabled**.

В ACL расширенные права доступа задаются в следующим формате:

```
extended permissions:
  enabled | disabled
    permit   режим  пользователь...
    deny     режим  пользователь...
    specify  режим  пользователь...
```

Каждая запись **permit**, **deny** и **specify** указывается на отдельной строке. Параметр *Mode* задается в формате **rwx** (при этом отсутствующие права доступа должны обозначаться дефисом (-)). Параметр *UserInfo* задается в формате `u:UserName`, `g:GroupName`, или в виде списка разделенных запятыми записей `u:UserName` и `g:GroupName`.

Примечание: Если в записи указано несколько имен пользователей, то ее нельзя применять для принятия решений о предоставлении доступа, поскольку процессу может соответствовать только один ИД пользователя.

Текстовое представление ACL типа AIXC

В следующем разделе рассматривается текстовое представление ACL типа AIXC:

```
Attributes:  { SUID | SGID | SVTX }
Base Permissions:
   owner(name): режим
   group(group): режим
   others: режим
Extended Permissions:
   enabled | disabled
     permit   режим  пользователь...
     deny     режим  пользователь...
     specify  режим  пользователь...
```

Двоичный формат ACL типа AIXC

Двоичный формат ACL типа AIXC определен в файле `/usr/include/sys/acl.h` и реализован в текущей версии AIX.

Пример ACL типа AIXC

Ниже приведен пример списка управления доступом типа 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:john, g:gateway, g:mail
        permit  rw-  g:account, g:finance
```

Ниже приведено описание записей ACL:

- Первая строка указывает, что включен бит **setuid**.
- Следующая строка, задающая базовые права доступа, необязательна.
- Следующие три строки определяют базовые права доступа. Имена владельца и группы в скобках приведены исключительно в информационных целях. Изменение этих имен не приведет к изменению владельца или группы файла. Изменить эти атрибуты файлов можно только с помощью команд **chown** и **chgrp**.
- Следующая строка, задающая расширенные права доступа, необязательна.
- В следующей строке указано, что перечисленные ниже расширенные права доступа включены.
- Следующие четыре строки содержат записи расширенных прав доступа. Первая запись расширенных прав доступа предоставляет пользователю *dhs* права доступа к файлу на чтение (r) и запись (w).
- Вторая запись расширенных прав доступа запрещает доступ на чтение (r) пользователю *chas* из группы *system*.
- Третья запись расширенных прав доступа указывает, что если пользователь *john* входит в состав групп *gateway* и *mail*, то ему должен быть разрешен доступ на чтение (r). Если пользователь *john* не входит хотя бы в одну из этих групп, то данные расширенные права доступа не применяются.
- Последняя запись расширенных прав доступа предоставляет всем пользователям, *одновременно* входящим в группы *account* и *finance*, права на чтение (r) и запись (w).

  Примечание: К процессу, запрашивающему доступ к защищенному объекту, может применяться несколько записей расширенных прав доступа, причем записи, запрещающие доступ, имеют более высокий приоритет, чем записи, разрешающие доступ.

Полная информация о синтаксисе приведена в описании команды **acledit** в руководстве *Справочник по командам*.

Списки управления доступом типа NFS4:

AIX также поддерживает списки управления доступом (ACL) типа NFS4.

Схема управления доступом, реализуемая с помощью списков ACL типа NFS4, описана в документе *RFC 3530, Протокол NFS версии 4*. В файловой системе JFS2 максимальный размер списков ACL типа NFS4 составляет 64 Кб.

Только клиент NFS V4 поддерживает NFS V4 ACL. Как Cachefs, так и Proxy, не поддерживают NFS V4 ACL.

Текстовое представление ACL типа NFS4

Текстовый список ACL типа NFS V4 представляет собой список записей управления доступом (ACE), каждая из которых расположена на отдельной строке. Запись ACE содержит четыре элемента в следующем формате:

```
IDENTITY    ACE_TYPE      ACE_MASK     ACE_FLAGS

где
IDENTITY => имеет формат 'тип-IDENTITY:(IDENTITY-имя или IDENTITY-ИД, либо IDENTITY-кто):'
         где
         тип-IDENTITY => Один из следующих типов субъектов:
                 u : пользователь
                 g : группа
                 s : специальная строка "кто"
```

```
                                (в качестве IDENTITY-кто должно быть
                                 указано ключевое слово)
                          IDENTITY-имя => имя пользователя/группы
                          IDENTITY-ИД   => ИД пользователя/группы
                          IDENTITY-кто  => особая строка "кто" (например, OWNER@, GROUP@, EVERYONE@)
ACE_TYPE => Один из следующих типов ACE:
                a : разрешить
                d : запретить
                l : предупредить
                u : контроль
ACE MASK => Один или несколько следующих ключей значений маски без разделителей:
                r : READ_DATA или LIST_DIRECTORY
                w : WRITE_DATA  или ADD_FILE
                p : APPEND_DATA  или ADD_SUBDIRECTORY
                R : READ_NAMED_ATTRS
                W : WRITE_NAMED_ATTRS
                x : EXECUTE  или 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 (необязательный элемент) => Один или несколько следующих ключей атрибутов без разделителей:
                fi : FILE_INHERIT
                di : DIRECTORY_INHERIT
                oi : INHERIT_ONLY
                ni : NO_PROPAGATE_INHERIT
                sf : SUCCESSFUL_ACCESS_ACE_FLAG
                ff : FAILED_ACCESS_ACE_FLAG
```

Примечание: Относительно значения ключа SYNCHRONIZE Ace_Mask, `s`, AIX не предпринимает никаких действий относительно данного значения ключа. Операционная система AIX сохраняет ключ значения `s`, но он не имеет никакого значения для AIX.

Когда ключу WRITE_OWNER Ace_Mask присвоено значение Ace_Type `allow`, пользователи могут указывать себя в качестве единственных владельцев файла.

Удаление файла зависит от двух ACE: записи DELETE удаляемого объекта и записи DELETE_CHILD его родительского каталога. Операционная система AIX предоставляет пользователям два режима действия. В *защищенном* режиме DELETE действует подобно AIXC ACL. В режиме *совместимость* DELETE ведет себя, как другие основные реализации NFS4 ACL. Для включения режима совместимости используйте команду **chdev** таким образом:

```
chdev -l sys0 -a nfs4_acl_compat=compatible
```

После выполнения команды **chdev** вам понадобится перезагрузить систему, чтобы чтобы изменения в настройках вступили в силу.

Если вы переключаете систему из режима в режим, то вам необходимо знать, что NFS4 ACL, созданные AIX в защищенном режиме, могут не приниматься другими платформами даже если система была возвращена в режим совместимости.

```
Пример:
        u:user1(aa@ibm.com):     a    rwp    fidi
        *s:(OWNER@):             d    x      dini          * Это строка - комментарий
        g:staff(jj@jj.com):      a    rx
        s:(GROUP@):              a    rwpx   fioi
        u:2:                     d    r     di             * Эта строка задана для пользователя (uid=2)
        g:7:                     a    ac    fi             * Эта строка отображает защиту группы (gid=7)
        s:(EVERYONE@):           a    rca   ni
```

Двоичный формат ACL типа NFS4

Двоичный формат ACL типа NFS4 определен в файле /usr/include/sys/acl.h и реализован в текущей версии AIX.

Пример ACL типа NFS4

В следующем примере показан список ACL типа NFS4 для каталога (например, j2eav2/d0):

```
s:(OWNER@):         a         rwpRWxDdo         difi         * Первая запись ACE
s:(OWNER@):         d         D                 difi         * Вторая запись ACE
s:(GROUP@):         d         x                 ni           * Третья запись ACE
s:(GROUP@):         a         rx                difi         * Четвертая запись ACE
s:(EVERYONE@):      a         c                 difi         * Пятая запись ACE
s:(EVERYONE@):      d         C                 difi         * Шестая запись ACE
u:user1:            a         wp                oi           * Седьмая запись ACE
g:grp1:             d         wp                             * Восьмая запись ACE
u:101:              a         C                              * Девятая запись ACE
g:100:              d         c                              * Десятая запись ACE
```

Ниже приведено описание записей ACL:

- Первая запись ACE указывает, что владелец имеет следующие права доступа к каталогу /j2eav2/d0 и всем его дочерним объектам, созданным после применения данного 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
- Вторая запись ACE указывает, что владелец не имеет права DELETE_CHILD (на удаление файлов или подкаталогов, созданных в /j2eav2), однако владелец все же может удалять их в силу наличия первой записи ACE, которая предоставляет владельцу право DELETE_CHILD.
- Третья запись ACE указывает, что всем членам группы данного объекта (/j2eav2/d0) не предоставлено право EXECUTE (=SEARCH_DIRECTORY), однако владелец имеет такое право в силу первой записи ACE. Эта запись ACE не наследуется дочерними объектами, поскольку задан флаг NO_PROPAGATE_INHERIT. Данная запись ACE применяется только к каталогу /j2eav2/d0, содержащимся в нем файлам и вложенным каталогам первого уровня.
- Четвертая запись ACE указывает, что всем участникам группы данного объекта (/j2eav2/d0) предоставлены права READ_DATA (= LIST_DIRECTORY) и EXECUTE (=SEARCH_DIRECTORY) на каталог /j2eav2/d0 и все его дочерние объекты. Однако в силу третьей записи ACE участники группы (за исключением владельца) не имеют прав EXECUTE (=SEARCH_DIRECTORY) на каталог /j2eav2/d0 и его прямых потомков.
- Пятая запись ACE указывает, что всем пользователям предоставлены права READ_ACL на каталог /j2eav2/d0 и все его дочерние объекты, созданные после применения данного ACL.
- Шестая запись ACE указывает, что всем пользователям отказано в правах WRITE_ACL на каталог /j2eav2/d0 и все его дочерние объекты. У владельцев файлов и каталогов со списками прав доступа типа NFS4 всегда есть права WRITE_ACL.
- Седьмая запись ACE указывает, что пользователю user1 предоставлены права WRITE_DATA (=ADD_FILE) и APPEND_DATA (= ADD_SUBDIRECTORY) на все дочерние объекты каталога /j2eav2/d0, но не на сам каталог /j2eav2/d0.

- Восьмая запись ACE указывает, что все участники группы grp1 не имеют прав WRITE_DATA (=ADD_FILE) и APPEND_DATA (=ADD_SUBDIRECTORY). В силу первой записи ACE данная запись не применяется для владельца, даже если он входит в состав группы grp1.
- Девятая запись ACE указывает, что пользователь с **UID 101** имеет права WRITE_ACL, однако в силу шестой записи ни один пользователь, кроме владельца, не имеет прав WRITE_ACL.
- Десятая запись ACE указывает, что ни один член группы с **GID 100** не имеет прав READ_ACL, но в силу пятой записи такие права им все же предоставляются.

Работа со списками управления доступом

Для просмотра и назначения ACL используются команды.

Создатели прикладных программ и другие разработчики подсистем могут использовать библиотечные интерфейсы ACL и функции преобразования ACL, описанные в этом разделе.

Команды администрирования ACL

Для работы со списками ACL объекта файловой системы можно использовать следующие команды:

aclget Записывает список ACL объекта файловой системы с именем *FileObject* в стандартный вывод или в файл вывода *outAclFile*, преобразовав список в формат, доступный для чтения.

aclput Создает список ACL для объекта файловой системы *FileObject*, используя данные из стандартного ввода или файла *inAclFile*.

acledit Открывает редактор для изменения списка ACL заданного объекта *FileObject*.

aclconvert
Преобразует список ACL в другой тип. Эта команда не работает в том случае, если преобразование не поддерживается.

aclgettypes
Получает типы списков ACL, которые поддерживаются файловой системой.

Библиотечные интерфейсы ACL

Библиотечные интерфейсы ACL работают как внешние интерфейсы прикладных программ, которым необходим доступ к спискам ACL. Прикладные программы (включая описанные выше общие команды администрирования ACL) не используют незадокументированные системные вызовы ACL напрямую. Вместо этого они используют универсальные системные вызовы, которые обращаются к загружаемым модулям для нужного типа ACL посредством библиотечных интерфейсов. С одной стороны, это ограждает создателей пользовательских прикладных программ от сложностей использования загружаемых модулей, а с другой стороны - уменьшает количество проблем двоичной совместимости с предыдущими версиями в будущих версиях AIX.

Системные вызовы используются в следующих библиотечных интерфейсах.

aclx_fget и aclx_get
Функции **aclx_get** и **aclx_fget** получают информацию об управлении доступом для объекта файловой системы и помещают эту информацию в область памяти, определенную **acl**. Объем и тип информации для **acl** определяются параметрами ***acl_sz** и ***acl_type**.

aclx_fput и aclx_put
Функции **aclx_put** и **aclx_fput** сохраняют информацию об управлении доступом, определенную в **acl**, для объекта файловой системы, указанного во вводе. Эти функции не преобразуют тип списка ACL; для выполнения такого преобразования необходимо явно вызвать функцию **aclx_convert**.

aclx_gettypes
Функция **aclx_gettypes** получает список типов ACL, которые поддерживаются данной файловой системой. В файловой системе может одновременно поддерживаться несколько типов списков ACL. Каждый объект файловой системы связан с одним из поддерживаемых типов ACL.

aclx_gettypeinfo
Функция **aclx_gettypeinfo** получает свойства и возможности типа ACL в файловой системе, указанной с помощью пути. Свойства ACL обычно возвращаются в виде структуры данных, тип которой уникален для каждого типа ACL. Структуры данных, которые используются для списков прав доступа AIXC и NFS4, будут описаны в отдельном документе.

aclx_print и aclx_printStr
Эти функции преобразуют список ACL, заданный в двоичном формате, в текстовый формат. Данные функции вызываются командами **aclget** и **acledit**.

aclx_scan и aclx_scanStr
Эти функции преобразуют список ACL, представленный в текстовом виде, в список в двоичном формате.

aclx_convert
Преобразует список ACL из одного типа в другой. Эта функция применяется для неявного преобразования такими командами, как **cp**, **mv** и **tar**.

Преобразование списков ACL

Списки ACL можно преобразовывать из одного типа в другой. Набор поддерживаемых типов ACL определяется конкретной физической файловой системой. Не все файловые системы поддерживают все типы ACL. Например, одна файловая система может поддерживать только списки прав доступа типа AIXC, а в другой файловой системе могут поддерживаться и списки типа AIXC, и списки типа NFS4. Списки ACL типа AIXC можно копировать из одной файловой системы в другую, но перед копированием списков ACL типа NFS их необходимо преобразовывать. При преобразовании ACL сохраняется максимальный объем информации об управлении доступом.

Примечание: Преобразование не создает точную копию исходного списка. Часть информации об управлении доступом может быть утеряна. Это следует иметь в виду при планировании преобразований ACL.

Для поддержки преобразования ACL в AIX предусмотрена следующая инфраструктура:

Библиотечные функции
Эти функции и среда ACL пользовательского уровня делают возможным преобразование ACL из одного типа в другой.

Команда aclconvert
Эта команда преобразует списки ACL.

Команды aclput и acledit
Эти команды используются для изменения типов ACL.

Команды cp и mv
Эти команды были адаптированы для поддержки нескольких типов ACL. При необходимости они выполняют внутреннее преобразование ACL.

Команда backup
Эта команда преобразует информацию списка ACL к известному типу и формату (ACL типа AIXC), если необходимо создать резервную копию в старом формате. Для восстановления списка ACL в исходном формате нужно указать опцию **-U**. Дополнительная информация приведена в разделе Резервное копирование.

Каждый тип ACL является уникальным, и значения масок прав доступа значительно отличаются для разных типов ACL. Алгоритмы преобразования являются приближенными, и их результат не совпадает с тем, что можно получить, выполнив преобразование вручную. В некоторых случаях преобразование не будет точным. Например, списки типа NFS4 нельзя правильно преобразовать в списки типа AIXC, поскольку списки NFS4 поддерживают 16 масок прав доступа и имеют функции наследования, которые не поддерживаются в

списках ACL типа AIXC). Если потеря информации об управлении доступом крайне нежелательна, то не используйте функции и интерфейсы преобразования ACL.

Примечание: Алгоритмы преобразования ACL являются закрытыми и могут изменяться.

S-биты и списки управления доступом

В этом разделе описывается работа с программами **setuid** и **setgid**, а также применение S-битов в списках управления доступом.

Применение программ setuid и setgid

Схема с разрешающим битом в большинстве случаев обеспечивает достаточно эффективное управление доступом к ресурсам. Для более строгого управления доступом служат программы **setuid** и **setgid**.

В операционной системе AIX для идентификации пользователя используются только идентификаторы uid и gid. Типы ACL, в которых применяются идентификаторы других форматов, преобразуются к модели идентификации AIX. Например, списки ACL типа NFS4 используют строковые идентификаторы пользователей вида пользователь@домен, и эти строки преобразуются в числовые значения UID и GID.

Как правило, программы запускаются с правами доступа пользователя и группы, соответствующими вызвавшему их пользователю. Владелец программы может связать с ней права доступа вызвавшего пользователя, установив в поле прав доступа бит **setuid** или **setgid**. Когда процесс вызывает такую программу, он получает права доступа ее владельца. Программа с установленным битом **setuid** при выполнении получает права доступа владельца, а программа с установленным битом **setgid** - права доступа группы. Поддерживается одновременная установка обоих типов.

Несмотря на то, что процессу предоставляются дополнительные права доступа, эти права контролируются выполняемой программой. Таким образом, установка битов **setuid** и **setgid** позволяет предоставлять права доступа косвенным образом. Программы играют роль защищенных подсистем, защищая таки образом права пользователей.

Применение таких программ повышает эффективность защиты, но если программа недостаточно надежна, то ее использование может привести к потере данных. В частности, программа не должна передавать управление пользователю, пока ей предоставлены права доступа владельца, так как в этом случае пользователю будут предоставлены все права доступа владельца этой программы.

Примечание: В целях обеспечения защиты система не поддерживает вызовы **setuid** и **setgid** из сценариев оболочки.

Применение S-битов к спискам ACL

ACL типа NFS4 не позволяют непосредственно работать с S-битами. В спецификации ACL типа NFS4 не указан способ их применения. В операционной системе AIX S-биты учитываются в ходе проверки прав доступа и дополняют информацию о них, связанную с ACL типа NFS4. Команда **chmod**, имеющаяся в операционной системе AIX, может использоваться для установки или сброса S-битов объектов файловой системы с такими ACL, как NFS4.

Административные права доступа

Операционная система предоставляет системным администраторам особые права доступа.

Системные права доступа зависят от ИД пользователя и группы. Пользователи, для которых действительный ИД пользователя или группы равен 0, считаются привилегированными.

Процессы, для которых ИД пользователя равен 0, называются процессами пользователя root-user. Эти процессы могут выполнять следующие операции:

- Чтение и запись любого объекта.

- Вызов любой системной функции.
- Управление системой с помощью программ **setuid-root**.

Управление системой можно осуществлять посредством прав доступа двух типов: команды **su** и программы**setuid-root**. Команда **su** позволяет всем вызываемым вами программам работать в режиме процесса пользователя root. Таким образом, команда **su** предоставляет гибкий, но не самый безопасный способ управления системой.

Программа **setuid-root** - это программа, владельцем которой является пользователь root и для которой установлен бит setuid. Программа **setuid-root** предоставляет обычным процессам возможность выполнять административные функции, не нарушая защиту: права доступа предоставляются программе, а не непосредственно пользователю. В некоторых случаях сложно реализовать все необходимые возможности только с помощью программ **setuid-root**, но такой подход обеспечивает наилучшую защиту системы.

Проверка прав доступа

Когда пользователь входит в систему с помощью команды **login** или **su**, процессу пользователя присваивается ИД пользователя и ИД группы той учетной записи, под управлением которой пользователь вошел в систему. Эти ИД определяют права доступа процесса.

Процесс с ИД пользователя 0 называется *процессом пользователя root*. Таким процессам разрешаются все способы доступа. Однако, если процесс пользователя root запрашивает доступ на выполнение, то выполнение разрешается лишь в том случае, если выполнение данного файла разрешено хотя бы одному пользователю.

Проверка прав доступа в списках ACL типа AIXC

Управление правами доступа возложено на владельца информационного ресурса. Защита ресурсов задается *битами прав доступа*, включенными в режим доступа к объекту. Биты прав доступа определяют права доступа, предоставленные владельцу объекта, группе объекта, а также всем остальным пользователям (класс `others`). Операционная система поддерживает три режима доступа, задаваемых независимо: чтение, запись и выполнение.

Для файлов, каталогов, именованных конвейеров и устройств (специальных файлов) проверка прав доступа выполняется следующим образом:

- Для каждой записи (ACE), указанной в списке управления доступом (ACL), список идентификаторов сравнивается с идентификаторами процесса. При обнаружении совпадения для процесса устанавливаются разрешения и запреты, определенные в данной записи. Итоговые права доступа определяются как логическое объединение всех разрешений и запретов, указанных в каждой из совпадающих записей ACL. Если запрашивающий процесс не соответствует ни одной из записей ACL, то для него устанавливаются разрешения и запреты, указанные в записи по умолчанию.
- Если запрошенный способ доступа разрешен (т.е. входит в логическое объединение всех разрешений) и не запрещен (т.е. не входит в логическое объединение всех запретов), то доступ предоставляется. В противном случае доступ запрещается.

Список идентификаторов ACL соответствует процессу в том случае, если все идентификаторы этого списка совпадают с действующим идентификатором соответствующего типа у запрашивающего процесса. Идентификатор типа USER должен совпадать с действующим ИД пользователя процесса, а идентификатор типа GROUP должен совпадать с действующим ИД группы процесса или с одним из дополнительных ИД групп. Допустим, что задана следующая запись списка прав доступа:

```
USER:fred, GROUP:philosophers, GROUP:software_programmer
```

такая запись будет соответствовать процессу с действующим ИД пользователя *fred* и следующим набором групп:

```
philosophers, philanthropists, software_programmer, doc_design
```

однако эта запись не будет соответствовать процессу с действующим ИД пользователя *fred* и следующим набором групп:

```
philosophers, iconoclasts, hardware_developer, graphic_design
```

Обратите внимание, что следующая запись списка прав доступа будет соответствовать обоим рассмотренным процессам:

```
USER:fred, GROUP:philosophers
```

Другими словами, список идентификаторов в записи ACL представляет собой список условий, разрешающих указанный способ доступа только в том случае, если выполнены все перечисленные условия.

Все проверки прав доступа к объектам выполняются на уровне системных вызовов при первом обращении к объекту. Поскольку обращение к объектам System V Interprocess Communication (SVIPC) выполняется без учета состояний этих объектов, то проверка выполняется при каждом обращении. Для объектов, имена которых соответствуют именам файловых систем, необходимо также обеспечить преобразование имени в фактическое расположение объекта. Имена преобразуются либо относительно (по отношению к рабочему каталогу процесса), либо абсолютно (по отношению к корневому каталогу процесса). Все операции преобразования имен начинаются с определения одной из этих точек отсчета.

Дискреционный механизм управления доступом обеспечивает эффективное управление доступом к информационным ресурсам и позволяет разделить защиту конфиденциальности и целостности информации. Эффективность механизма управления доступом от имени владельца определяется исключительно тем, как пользователи используют этот механизм. Все пользователи должны понимать принципы настройки, проверки и предоставления прав доступа.

Проверка прав доступа в списках ACL типа NFS4

Любой пользователь, имеющий право на ЗАПИСЬ_ACL, может управлять правами доступа. Владелец информационного ресурса всегда имеет права WRITE_ACL. Для файлов и каталогов со списками ACL типа NFS4 права доступа предоставляются следующим образом:

- Список записей ACE просматривается сверху вниз. Для дальнейшей обработки из него отбираются только те записи, которые содержат параметр "who" (т.е. субъект), который соответствует инициатору. Идентификационные данные инициатора не проверяются при обработке записи ACE, содержащей специальную строку who EVERYONE@.
- Записи ACE обрабатываются до тех пор, пока не будет получено подтверждение, что разрешены все биты доступа инициатора. После получения подтверждения того, что бит разрешен, он исключается из рассмотрения при обработке дальнейших записей ACE.
- При обнаружении запрета для одного из битов доступа инициатора, доступ не предоставляется, а дальнейшие записи ACE не обрабатываются.
- Если после обработки всех записей ACL не было получено подтверждение того, что разрешены все биты доступа инициатора, доступ будет запрещен.

Если по результатам обработки записей ACE запрашиваемый тип доступа должен быть запрещен, однако пользователь, запросивший доступ, является администратором или пользователем root, то доступ обычно предоставляется. Обратите внимание, что владельцу объекта всегда предоставляется доступ типа READ_ACL, WRITE_ACL, READ_ATTRIBUTES и WRITE_ATTRIBUTES. Дополнительная информация об алгоритме проверки прав доступа приведена в разделе “Списки управления доступом типа NFS4” на стр. 127.

Устранение неполадок списков управления доступом

В этом разделе обсуждаются вопросы по устранению неполадок в списках управления доступом (ACL).

Ошибки списка управления доступом типа NFS4 для объекта приложения

С помощью кода возврата или утилиты трассировки найдите ошибки, допущенные при составлении списка ACL типа NFS4 для объекта, например для файла или каталога. В обоих случаях для обнаружения причины

ошибки следует использовать команды **aclput** и **acledit**.

Устранение неполадки с помощью кода возврата

Для просмотра кода возврата введите команду `echo $?` после выполнения команды **aclput**. Ниже перечислены все возможные коды возврата:

22 (EINVAL, определен в файле /usr/include/sys/errno.h)

Возможны следующие причины появления этого кода:

- Недопустимый текстовый формат в каком-либо из 4 полей.
- Размер входного списка ACL типа NFS4 превышает 64 Кб.
- Список ACL применяется к файлу, который уже имеет по крайней мере одну запись ACE с установленной маской `w` (`WRITE_DATA`), но не `p` (`APPEND_DATA`), либо `p` (`APPEND_DATA`), но не `w` (`WRITE_DATA`).
- Список ACL применяется к каталогу, который уже имеет по крайней мере одну запись ACE с маской `w` (`WRITE_DATA`), но не `p` (`APPEND_DATA`), либо `p` (`APPEND_DATA`), но не `w` (`WRITE_DATA`), и флагом `fi` (`FILE_INHERIT`).
- Существует по крайней мере одна запись ACE со значением OWNER@, указанным в виде специальной строки **who** (**Identity**), и одна или несколько масок ACE `c` (`READ_ACL`), `C` (`WRITE_ACL`), `a` (`READ_ATTRIBUTE`) и `A` (`WRITE_ATTRIBUTE`) запрещены для ACE типа `d`.

124 (ENOTSUP, определен в /usr/include/sys/errno.h)

Возможны следующие причины появления этого кода:

- Специальная строка who не совпадает ни с одним из трех значений (`OWNER@`, `GROUP@` или `EVERYONE@`) в одной из записей ACE.
- Существует по крайней мере одна запись типа `u` (`AUDIT`) или `l` (`ALARM`).

13 (EACCES, определен в /usr/include/sys/errno.h)

Возможны следующие причины появления этого кода:

- Нет прав на чтение файла ввода, содержащего записи ACE NFS4.
- Нет прав на поиск в родительском каталоге целевого объекта, так как по отношению к этому каталогу не предоставлены права доступа `x` (`ВЫПОЛНЕНИЕ`).
- Нет прав на запись или изменение ACL. Если объект уже связан со списком ACL типа NFS4, убедитесь, что у вас есть права доступа, соответствующие маске ACE `C` (`WRITE_ACL`).

Устранение неполадок с помощью утилиты трассировки

Для определения причины неполадки можно создать отчет о трассировке. В следующем сценарии показано, как использовать трассировку для определения причины неполадки в случае использования списка ACL типа NFS4. Предположим, что имеется файл `/j2v2/file1` со следующим списком ACL типа NFS4:

```
s:(EVERYONE@):  a       acC
```

Пусть в файле ввода `input_acl_file` содержится следующий список ACL:

```
s:(EVERYONE@):  a       rwxacC
```

Для устранения неполадки с помощью утилиты трассировки выполните следующие действия:

1. Запустите трассировку, **aclput** и **trcrpt**, используя следующие команды: **aclput** and **trcrpt** using the following commands:

   ```
   $ trace -j  478 -o trc.raw
   $->!aclput -i input_acl_file -t NFS4 /j2v2/file1
   $ ->quit
   $ trcrpt trc.raw > trc.rpt
   ```

2. Проанализируйте отчет о трассировке. При применении списка ACL к файлу или каталогу проверяется наличие прав на запись или изменение списка ACL, а затем применяется список ACL. В файле содержатся строки следующего вида:

```
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
```

   Содержимое второй строки, `chk_access exit`, означает наличие разрешения на запись ACL (`rc = 0`). Содержимое четвертой строки, `validate_acl`, и пятой, `set_acl exit`, обозначает, что список ACL не применен (`rc=22` означает **EINVAL**). Содержимое четвертой строки, `validate_acl`, указывает, что ошибка содержится в первой строке ACE (`ace_cnt=1`). Если рассмотреть первую запись ACE, `s:(EVERYONE@): a rwxacC)`, то можно обнаружить, что в ней отсутствует маска права доступа **p**. Для применения списка прав доступа в дополнение к **w** необходимо указать **p**.

Устранение неполадок прав доступа

При выполнении операции файловой системы (такой как чтение или запись) для объекта, с которым связан ACL типа NFS4, может возникнуть сбой. Обычно в этом случае выдается сообщение об ошибке, однако это сообщение может содержать недостаточно информации для выявления ошибки прав доступа. Для обнаружения ошибок прав доступа можно использовать утилиту трассировки. Например, пусть имеется файл `/j2v2/file2` со следующим списком ACL NFS4:

```
s:(EVERYONE@):  a       rwpx
```

Следующая команда выдает сообщение "Доступ запрещен":

```
ls -l /j2v2/file2
```

Для обнаружения причины ошибки выполните следующее:

1. Запустите трассировку, `ls -l /j2v2/file2`, и trcrpt с помощью следующих команд:

```
$ trace -j  478 -o trc.raw
$->!ls -l /j2v2/file2
$ ->quit
$ trcrpt trc.raw > trc.rpt
```

2. Проанализируйте отчет о трассировке. В файле содержатся строки следующего вида:

```
478  xxx  xxx   ACL ENGINE: chk_access entry: type=NFS4 obj_mode=33587711 size=68  ps=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
```

   Третья строка указывает, что доступ запрещен для `маски прав доступа = 128` (**0x80**), которая дает только права `READ_ATTRIBUTES` (см. файл `/usr/include/sys/acl.h`).

Обзор подсистемы контроля

Подсистема контроля позволяет системному администратору сохранять данные, относящиеся к защите системы. В дальнейшем, проанализировав эти данные, он сможет обнаружить реальные и потенциальные нарушения стратегии защиты системы.

Подсистема контроля

У подсистемы контроля есть функции отслеживания, контроля и обработки событий.

- “Обнаружение контрольных событий” на стр. 137
- “Сбор данных о событии” на стр. 137
- “Обработка данных контрольного журнала” на стр. 137

Системный администратор может настраивать любую из этих функций.

Обнаружение контрольных событий

В отслеживании событий принимает участие как ядро защищенной компьютерной базы (TCB) (режим супервизора), так и защищенные программы (пользовательский режим). Отслеживается любое событие в системе, имеющее отношение к защите. К таковым относится любое изменение состояния защиты, а также любое потенциальное или фактическое нарушение правил доступа к системе и/или стратегий защиты и учета ресурсов. Программы и модули ядра, обнаруживающие отслеживаемые события, должны сообщать о таких событиях системной программе ведения протокола контроля, представляющей собой часть ядра. Обратиться к этой программе можно либо с помощью подпрограммы (в пользовательском режиме), либо путем вызова процедуры ядра (в режиме супервизора). Информация о событии включает его имя, сведения об успешном или неудачном завершении, а также другую дополнительную информацию, имеющую отношение к системе защиты.

Настройка программы отслеживания событий заключается в ее включении/выключении и выборе наборов отслеживаемых событий для различных пользователей. Для включения функции отслеживания событий применяется команда **audit**, позволяющая включать и выключать подсистему контроля. Список пользователей и событий, обрабатываемых подсистемой контроля, хранится в файле `/etc/security/audit/config`

Сбор данных о событии

При сборе данных выполняется регистрация выбранных отслеживаемых событий. За эту функцию отвечает программа ведения протокола ядра, обеспечивающая как работу с системными вызовами, так и интерфейс вызова внутренних процедур ядра, регистрирующих события.

Программа ведения протокола контроля формирует полную запись протокола контроля, включающую заголовок с информацией, общей для всех событий (имя события, имя пользователя, время и состояние события), а также блок информации с данными, характерными для конкретного типа событий. Программа ведения протокола добавляет каждую последующую запись к контрольному журналу ядра. Запись контрольного журнала можно вести в любом из двух (или в обоих сразу) режимах:

Режим лотка
: Журнал записывается в файлы, которые затем сохраняются.

Режим потока
: Журнал записывается в замкнутый буфер, данные из которого синхронно считываются с помощью псевдоустройства контроля. Режим потока позволяет получать данные незамедлительно.

Можно настроить обе части процесса сбора данных - как запись события, так и обработку контрольного журнала. Вы можете записывать события, относящиеся к конкретному пользователю. Для каждого пользователя определен собственный набор отслеживаемых событий, которые и записываются в контрольный журнал. В системе предусмотрена возможность независимой настройки каждого режима обработки контрольного журнала, и системный администратор может применить наиболее подходящий для данной среды способ. Кроме того, можно настроить режим лотка таким образом, чтобы в случае уменьшения объема доступного дискового пространства в файловой системе, предназначенной для хранения журнала, выдавалось предупреждение.

Обработка данных контрольного журнала

Операционная система обеспечивает несколько вариантов обработки контрольного журнала ядра. В режиме лотка перед отправкой в архив контрольный журнал может быть сжат, обработан с помощью фильтров или отформатирован для вывода. При сжатии применяется метод Хаффмана. Фильтрация выполняется путем выбора отдельных записей контроля (с помощью команды **auditselect** и операторов, аналогичных операторам SQL). Она применяется как для выборочного просмотра записей, так и для выборочного

сохранения контрольного следа. Форматирование записей контрольного следа применяется для их проверки и просмотра, создания периодических отчетов о состоянии защиты, а также для печати копии контрольного следа.

За контрольным следом, созданным в режиме потока, можно следить постоянно, немедленно реагируя на все потенциально опасные ситуации. Этой функцией управляют независимые программы-демоны, которые могут фильтровать данные обоих режимов, хотя некоторые из них предназначены для какого-либо одного режима.

Настройка подсистемы контроля

Переменная глобального состояния подсистемы контроля показывает, включена ли данная подсистема. Кроме того, у каждого процесса есть локальная переменная состояния, значение которой показывает, следует ли подсистеме контроля записывать данные об этом процессе.

Обе переменные показывают, были ли обнаружены какие-либо события модулями и программами защищенной компьютерной базы (TCB). Если отключить для определенного процесса контроль TCB, то этот процесс сможет выполнять контроль самостоятельно, не влияя при этом на стратегию отчетности системы. Самостоятельный контроль защищенной программы обеспечивает более эффективный сбор данных.

Сбор информации подсистемы контроля

Сбор данных включает в себя обнаружение событий и создание контрольного журнала ядра. Все необходимые функции выполняет процедура ядра, с помощью которой компоненты TCB обнаруживают контролируемые события, и интерфейсы настройки, через которые подсистема контроля управляет процедурой занесения событий в протокол.

Ведение протокола контроля

Занесение отслеживаемых событий в протокол выполняется с помощью одного из интерфейсов: пользовательского режима или режима супервизора. Пользовательский компонент TCB вызывает для этого подпрограмму **auditlog** или **auditwrite**, а компонент TCB, работающий в режиме супервизора, использует набор вызовов процедур ядра.

Программа занесения в протокол добавляет контрольный заголовок к информации о каждом событии. В заголовке указано имя пользователя и процесса отслеживаемого события, а также время, когда оно произошло. Обнаружившая событие программа сообщает следующие данные: тип события, код возврата или его состояние. Иногда передается дополнительная информация о событии (контрольный след события). Эта дополнительная информация включает в себя имена объектов (например, имена файлов, доступ к которым был запрещен, или имена терминалов tty, на которых произошел сбой при попытке входа в систему), параметры подпрограмм и прочие данные.

Для определения событий как правило используются символьные имена, а не числовые. При отсутствии четкой схемы регистрации событий это позволяет снизить вероятность конфликтов имен. Поскольку в число контролируемых объектов входят и подпрограммы, а в расширяемом определении ядра не определены фиксированные номера коммутируемых виртуальных контуров (SVC), числовые идентификаторы событий применять достаточно сложно. Это связано с тем, что правила нумерации событий приходилось бы пересматривать при каждом расширении или переопределении интерфейса ядра.

Формат контрольной записи

Контрольная запись состоит из общего заголовка и прикрепленного к нему контрольного журнала, который содержит информацию, описывающую конкретное событие. Структуры заголовка определены в файле `/usr/include/sys/audit.h`. Формат записи в контрольном журнале зависит от типа события. Форматы для основных событий определены в файле `/etc/security/audit/events`.

Сбор данных для контрольного заголовка обычно выполняет процедура занесения в протокол, обеспечивающая большую точность, а данные контрольного следа предоставляет программа, обнаружившая событие. Программа занесения в протокол не знакома со структурой и семантикой представления данных контрольного следа. Например, обнаружив ошибку при входе в систему, команда **login** определяет имя терминала, фиксирует данные об этом событии, а затем с помощью подпрограммы **auditlog** записывает эту информацию в контрольный журнал. Программа создания протокола ядра записывает специальную информацию об объекте (ИД пользователя и процесса, время) в заголовок, который добавляется к прочим данным. Единственное, что записывает сама вызывающая программа - это расположенное в заголовке имя события и поле результата.

Настройка ведения протокола контроля

Задачей программы занесения данных в протокол является составление полной контрольной записи. Вы должны выбрать, какие события необходимо заносить в протокол.

Выбор контрольных событий

При выборе контрольных событий возможны следующие варианты:

Отслеживание событий на уровне процесса

Для более эффективного выбора событий, связанных с процессом, системный администратор может определить классы контроля. Класс контроля представляет собой подмножество основных подлежащих контролю событий. Разбиение на классы облегчает логическую группировку основных контрольных событий.

Для каждого пользователя системный администратор задает набор классов контроля, определяющий основные события для занесения в протокол. Каждый процесс, запущенный пользователем, относится к одному из его контрольных классов.

Отслеживание событий на уровне объекта

Операционная система позволяет отслеживать все обращения к объекту с заданным именем. Иными словами, она обеспечивает контроль за отдельными объектами (обычно такими объектами являются файлы). Контроль за объектами по их имени позволяет избежать необходимости следить за всеми файлами, для того чтобы проконтролировать только некоторые из них. Кроме того, можно задать режим контроля, что позволяет записывать только информацию об определенных способах доступа (чтение/запись/выполнение).

Режимы контрольного журнала ядра

Место записи контрольного журнала ядра зависит от режимов ведения протокола ядра (режим лотка и режим потока). В режиме лотка программе ведения протокола контроля ядра перед запуском контроля необходимо передать по крайней мере один дескриптор файла, в который будут добавляться записи протокола.

В режиме лотка контрольные данные записываются в сменяющиеся файлы. В момент запуска процесса контроля в ядро передаются два дескриптора файлов и рекомендуемое значение максимального размера файла-приемника. Вызывающий процесс приостанавливается, и начинается запись контрольных данных в первый файл. Когда размер первого файла-приемника достигает максимального значения, процесс переключается на второй файл (если он указан верно), а вызывающий процесс вновь активируется. Ядро продолжает запись во второй файл до тех пор, пока он не будет вызван снова с указанием следующего дескриптора файла. Если в этот момент второй файл полон, то процесс переключится обратно на первый файл-приемник, и вызывающий процесс немедленно возобновится. В противном случае, процесс приостановится, и ядро продолжит записывать данные во второй файл до тех пор, пока он не заполнится. Таким образом обработка продолжается вплоть до ее выключения. На следующем рисунке проиллюстрирован контроль в режиме лотка:

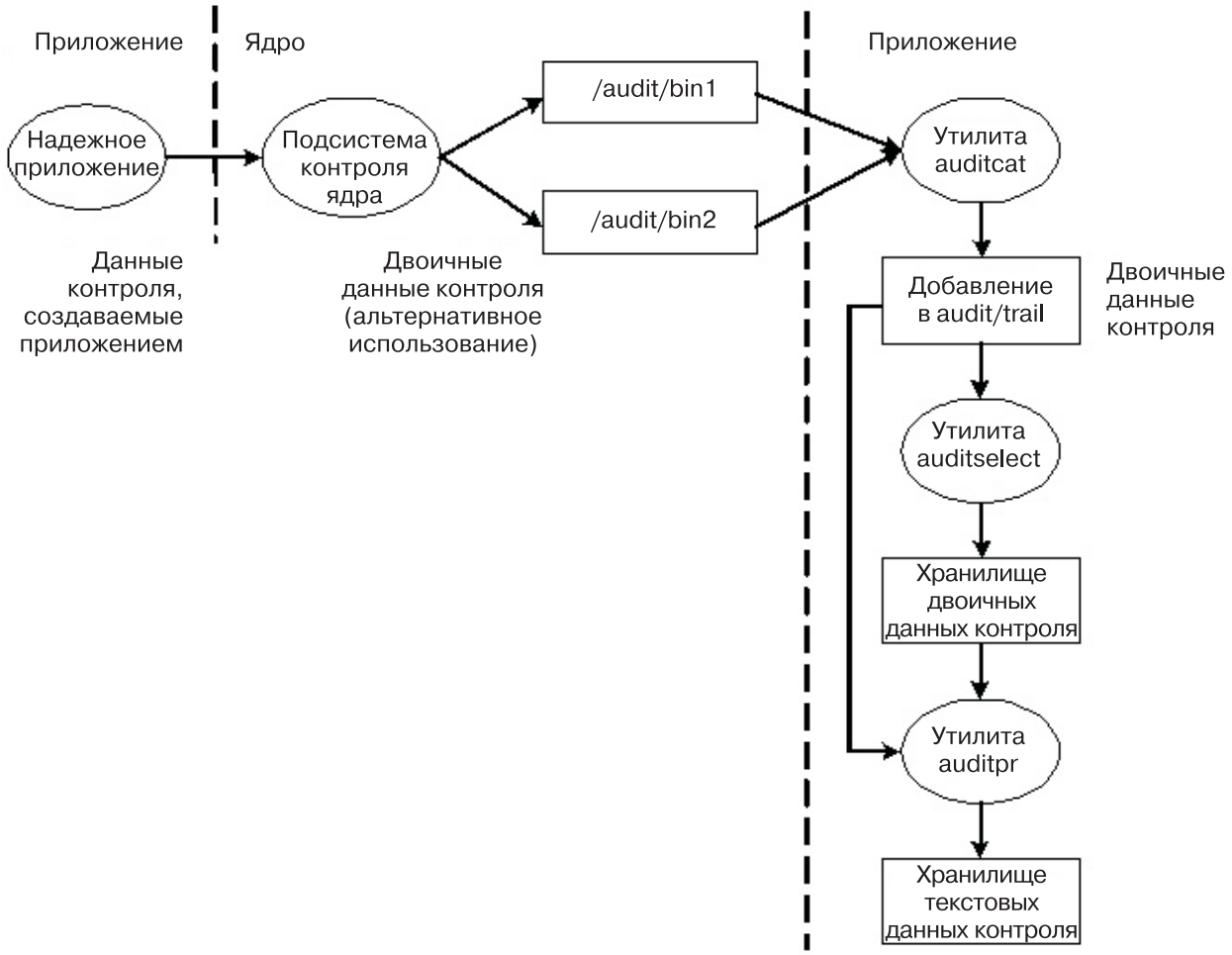


Рисунок 1. Контроль в режиме лотка. Описан процесс контроля в режиме лотка

Механизм лотка позволяет обеспечить постоянное наличие файла-приемника при обработке контрольных записей. При переключении на другой файл-приемник содержимое первого файла перемещается в файл `trace`. Когда возникнет необходимость снова переключиться на первый файл-приемник, этот файл будет доступен для записи. Такой подход позволяет разделить процессы записи и анализа данных. Как правило, для чтения данных из файла-приемника, в который информация в настоящий момент не записывается ядром, применяется программа **auditcat**. Для того чтобы в системе всегда было достаточно места для записи контрольного журнала (вывода программы **auditcat**), в файле `/etc/security/audit/config` необходимо указать параметр *freespace*. Если число свободных 512-байтовых блоков в системе становится меньше указанного здесь значения, то формируется сообщение syslog.

Если контроль включен, то параметру *binmode* в разделе `start` файла`/etc/security/audit/config` следует присвоить значение `panic`. Параметру *freespace* в разделе `bin` должно быть присвоено значение, соответствующее не менее чем 25 процентам общего дискового пространства, выделенного для сохранения контрольного следа. Параметрам *bytethreshold* и *binsize* должно быть присвоено значение 65536 байт.

В режиме потока ядро добавляет записи в замкнутый буфер. Когда буфер заполняется, ядро просто начинает снова записывать данные в его начало. Процессы ядра считывают информацию с помощью псевдоустройства `/dev/audit`. В момент открытия процессом устройства для процесса создается новый канал. События, считываемые на данном канале, можно определить в виде списка классов контроля. На следующем рисунке проиллюстрирован контроль в режиме потока:

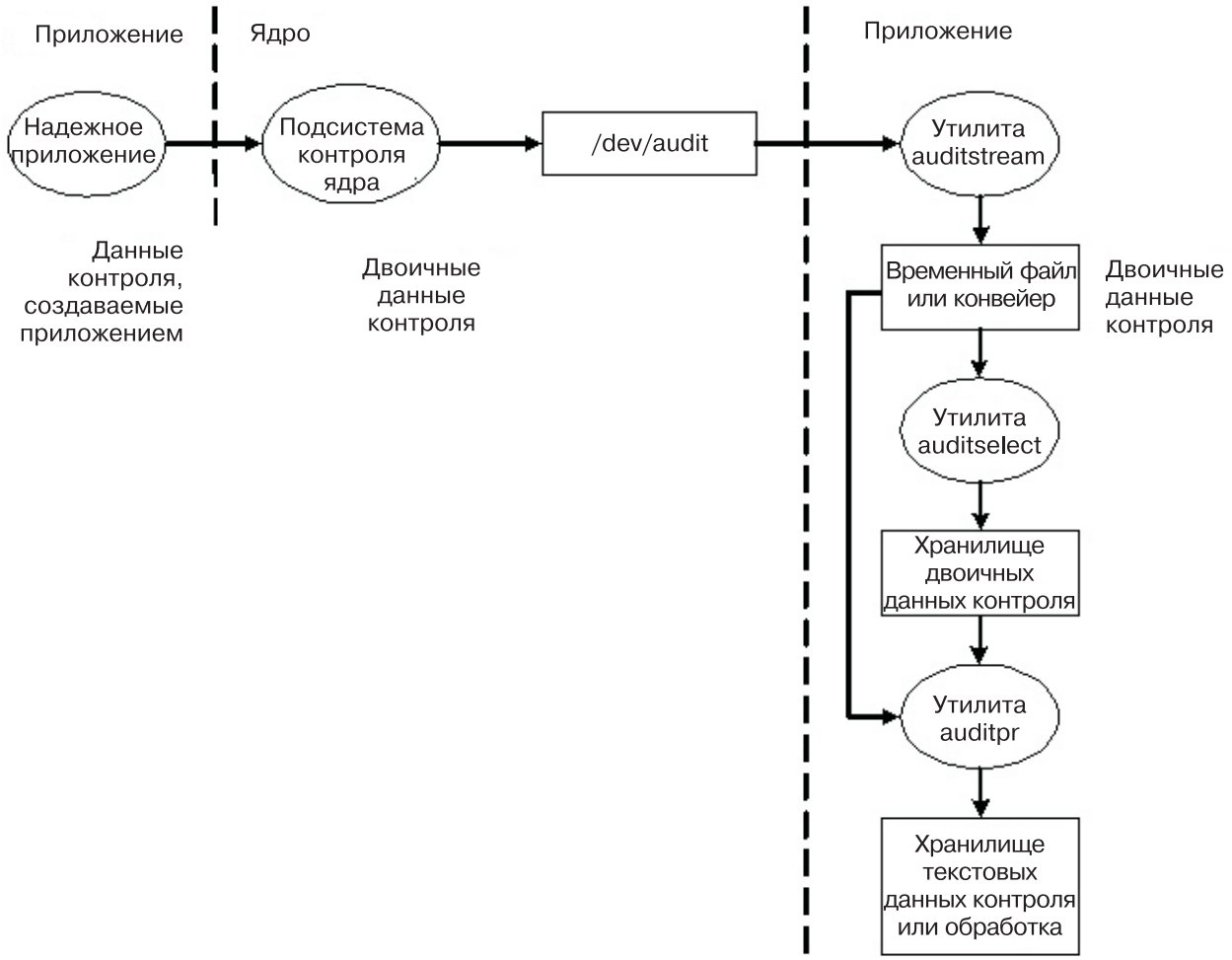


Рисунок 2. Контроль в режиме потока. Описан процесс контроля в режиме потока

Потоковый режим предназначен для тех случаев, когда периодически требуется просматривать контрольный след, например, в случае контроля защиты в реальном времени. Еще одна область применения этого режима - немедленная запись данных контрольного следа, без необходимости вмешательства и нарушения контрольного следа (это может произойти, если след хранится на каком-либо носителе).

Еще один способ применения потокового режима заключается в передачи потока контрольных данных программе, которая сохраняет данные контроля в удаленной системе, обеспечивая оперативную централизованную обработку и защиту от фальсификации данных контроля на исходном хосте.

Обработка контрольных записей

Для обработки контрольных записей в режиме лотка или потока применяются команды **auditselect**, **auditpr** и **auditmerge**. Все эти утилиты работают по принципу фильтров, поэтому их можно использовать в конвейерах, что особенно удобно в режиме потока.

auditselect

С помощью операторов, аналогичных SQL-like, позволяет выбирать отдельные контрольные записи. Например, для выбора только событий **exec()**, связанных с пользователем *afx*, введите:

```
auditselect -e "login==afx && event==PROC_Execute"
```

auditpr

Позволяет преобразовать двоичные контрольные записи в формат, удобный для пользователей.

Объем показанной информации зависит от заданных флагов. Для включения всей доступной информации необходимо ввести следующую команду **auditpr**:

```
auditpr -v -hhelrtRpPTc
```

Если указан флаг **-v**, то помимо стандартной информации, сохраняемой ядром для каждого события, отображается также контрольный журнал, сой для каждого события (см. файл `/etc/security/audit/events`).

auditmerge

Позволяет объединять двоичные контрольные журналы. Эта команда особенно полезна при объединении контрольных журналов из нескольких систем. Команда **auditmerge** получает в качестве исходных параметров имена контрольных журналов и направляет объединенный двоичный контрольный журнал в стандартный поток вывода. Для работы с полученной информацией можно воспользоваться командой **auditpr**. Например, можно ввести следующее сочетание команд **auditmerge** и **auditptr**:

```
auditmerge trail.system1 trail.system2 | auditpr -v -hhelrRtpc
```

Применение подсистемы контроля для оперативной проверки защиты:

Для отслеживания отдельной подозрительной программы без настройки подсистемы контроля можно воспользоваться командой **watch**. Эта команда сохраняет полный или частичный список событий, сформированных указанной программой.

Например, чтобы посмотреть все события **FILE_Open** программы **vi /etc/hosts**, введите команду:

```
watch -eFILE_Open -o /tmp/vi.watch vi /etc/hosts
```

В файле `/tmp/vi.watch` будут перечислены все события **FILE_Open** из сеанса редактирования.

Выбор событий

При выборе событий следует придерживаться баланса между недостаточным и чрезмерным количеством сведений.

События, которые система будет отслеживать, а также степень тщательности контроля определяются набором подлежащих контролю событий. Как уже было сказано ранее, в этот набор должны входить все события, имеющие отношение к защите системы. При создании определения события, подлежащего контролю, необходимо избежать как излишне подробного описания (которое приведет к сбору слишком большого объема избыточных данных), так и слишком краткого описания (которое затруднит для системного администратора понимание полученной информации). Описания сходных обнаруженных событий также очень похожи между собой. Далее под *обнаруженным событием* будет пониматься отдельный экземпляр контролируемого события, поскольку одно и то же событие может быть зафиксировано несколько раз. Основной принцип заключается в том, что обнаруженные события со сходными свойствами (в отношении защиты системы) считаются одним и тем же контролируемым событием. В следующем списке приведена классификация событий стратегии защиты:

- События субъекта
  - Создание процесса
  - Удаление процесса
  - Установка атрибутов защиты: ИД пользователя или группы
  - Группа процессов, управление терминалом
- События объекта
  - Создание объекта
  - Удаление объекта
  - Открытие объекта (включая процессы в качестве объектов)
  - Закрытие объекта (включая процессы в качестве объектов)

  - Установка атрибутов защиты объекта: владельца, группы и ACL
- События импорта/экспорта
  - Импорт или экспорт объекта
- События учета
  - Добавление пользователя, изменение пользовательских атрибутов в базе данных паролей
  - Добавление группы, изменение атрибутов группы в базе данных групп
  - Вход пользователя в систему
  - Выход пользователя из системы
  - Изменение идентификационных данных пользователя
  - Настройка терминала защищенного пути
  - Настройка идентификации
  - Управление контролем: выбор событий и контрольных следов, включение и выключение контроля, определение контрольных классов пользователей
- Общие события администрирования системы
  - Использование привилегий
  - Настройка файловых систем
  - Определение и настройка устройств
  - Определение параметров настройки системы
  - Выполнение IPL или завершение работы системы
  - Настройка RAS
  - Прочие задачи настройки системы
  - Запуск подсистемы контроля
  - Остановка подсистемы контроля
  - Отправка запросов в подсистеме контроля
  - Перезапуск подсистемы контроля
- Нарушения защиты (потенциальные)
  - Случаи отказа в предоставлении доступа
  - Сбои при проверке привилегий
  - Сбои и системные ошибки, обнаруженные программами диагностики
  - Попытки изменения TCB

События контроля:

Событие контроля - это любое событие в системе, имеющее отношение к защите. К таковым относится любое изменение состояния защиты, а также любое потенциальное или фактическое нарушение правил доступа к системе и/или стратегий защиты и учета ресурсов. Программы и модули ядра, обнаруживающие события контроля, сообщают о них системной программе ведения протокола контроля, представляющей собой часть ядра. Обратиться к этой программе можно либо с помощью подпрограммы (в пользовательском режиме), либо путем вызова процедуры ядра (в режиме супервизора). Информация о событии контроля включает его имя, сведения об успешном или неудачном завершении, а также другую дополнительную информацию, имеющую отношение к системе защиты.

Для того чтобы начать контроль какого-либо действия, необходимо выяснить, какая программа или процесс инициализируют контрольное событие, и указать это событие в файле `/etc/security/audit/events` вашей системы. Процесс присвоения контрольных событий пользователям можно упростить, объединив похожие события в контрольные классы. Их определения находятся в разделе классов файла `/etc/security/audit/config`.

В следующей таблице перечислены отдельные общие события контроля, возникающие в операционной системе AIX:

Таблица 11. События контроля

Вызов пользователя или системный вызов	Событие контроля	Описание
fork	PROC_Create	Указывает, что создается процесс.
exit	PROC_Delete	Указывает, что вызывающий процесс завершен.
exec	PROC_Execute	Запускает новую программу.
setuidx	PROC_RealUID	Задает ИД пользователя процесса.
	PROC_AuditID	
	PROC_SetUserIDs	
setgidx	PROC_RealGID	Задает ИД группы процессов.
accessx	FILE_Accessx	Определяет, доступен ли файл.
statacl	FILE_StatAcl	Извлекает информацию об управлении доступом к файлу.
revoke	FILE_Revoke	Аннулирует доступ к файлу для всех процессов.
frevoke	FILE_Frevoke	Аннулирует доступ к файлу для других процессов.
usrinfo	PROC_Environ	Изменяет часть информации о пользователе.
sigaction	PROC_SetSignal	Задает действие, которое выполняется в ответ на доставку в процесс, вызвавший эту процедуру, конкретного сигнала.
setrlimit	PROC_Limits	Управляет использованием максимальных системных ресурсов.
nice	PROC_SetPri	Задает использование функции `nice`.
setpri	PROC_Setpri	Задает фиксированный приоритет для процессов.
setpriv	PROC_Privilege	Изменяет один или несколько векторов прав доступа для процессов.
settimer	PROC_Settimer	Задает текущее значение указанного таймера уровня системы.
adjtime	PROC_Adjtime	Изменяет системные часы.
ptrace	PROC_Debug	Обеспечивает трассировку другого процесса.
kill	PROC_Kill	Передает сигнал в процесс или группу процессов.
setpgid	PROC_setpgid	Задает ИД группы процессов.
ld_loadmodule	PROC_Load	Загружает новый модуль объектов в адресное пространство процесса.
	PROC_LoadError	Указывает на сбой загрузки объекта.
setgroups	PROC_SetGroups	Изменяет параллельный набор групп процессов.
sysconfig	PROC_Sysconfig	Получает действие над ядром или конфигурацией системы.
audit	AUD_It	Запускает и останавливает операцию контроля. Кроме того, запрашивает состояние контроля.
auditbin	AUD_Bin_Def	Изменяет системный вызов `auditbin`.
auditevents	AUD_Events	Изменяет события.
auditobj	AUD_Objects	Изменяет системный вызов `auditobj`.

Таблица 11. События контроля (продолжение)

Вызов пользователя или системный вызов	Событие контроля	Описание
auditproc	AUD_Proc	Получает или задает состояние контроля процесса.
acct	ACCT_Disable	Выключает учет ресурсов системы.
	ACCT_Enable	Включает учет ресурсов системы.
open и create	FILE_Open	Вызывает процедуру **open**.
read	FILE_Read	Считывает данные из дескриптора файла.
write	FILE_Write	Записывает данные в дескриптор файла.
close	FILE_Close	Закрывает открытый дескриптор файла.
link	FILE_Link	Создает новую запись каталога для объекта файловой системы.
unlink	FILE_Unlink	Удаляет объект файловой системы.
rename	FILE_Rename	Изменяет имя объекта файловой системы.
chown	FILE_Owner	Изменяет принадлежность файла.
chmod	FILE_Mode	Изменяет режим файла.
fchmod	FILE_Fchmod	Изменяет права доступа к дескриптору файла.
fchown	FILE_Fchown	Изменяет принадлежность дескриптора файла.
truncate	FILE_Truncate	Изменяет длину обычных файлов или объекта общей памяти.
symlink	FILE_Symlink	Создает символьную ссылку.
pipe	FILE_Pipe	Создает канал без имени.
mknod	FILE_Mknod	Создает специальный файл устройства или специальный файл FIFO.
fcntl	FILE_Dupfd	Создается копию дескриптора файла.
fscntl	FS_Extend	Расширяет файловую систему.
mount	FS_Mount	Подключает файловую систему к именованному каталогу.
umount	FS_Umount	Отключает смонтированную файловую систему.
chacl	FILE_Acl	Изменяет список управления доступом (ACL) файла.
fchacl	FILE_Facl	Изменяет ACL дескриптора файла.
chpriv	FILE_Privilege	Задает список управления привилегиями (PCL) для пути к файлу.
	FILE_Chpriv	Изменяет PCL.
	FILE_Fchpriv	Изменяет PCL дескриптора файла.
chdir	FS_Chdir	Изменяет текущий рабочий каталог.
fchdir	FS_Fchdir	Изменяет текущий рабочий каталог с помощью помощи дескриптора файла.
chroot	FS_Chroot	Изменяет значение корневого каталога (/) для текущего процесса.
rmdir	FS_Rmdir	Удаляет объект каталога.
mkdir	FS_Mkdir	Создает каталог.
utimes	FILE_Utimes	Вызывает процедуру **utimes**.
stat	FILE_Stat	Вызывает процедуру **stat**.
msgget	MSG_Create	Создает очередь сообщений.
msgrcv	MSG_Read	Получает сообщение из очереди сообщений.

Таблица 11. События контроля (продолжение)

Вызов пользователя или системный вызов	Событие контроля	Описание
msgsnd	MSG_Write	Отправляет сообщение в очередь сообщений.
msgctl	MSG_Delete	Удаляет очередь сообщений.
	MSG_Owner	Изменяет принадлежность очереди сообщений и права доступа к ней.
	MSG_Mode	Запрашивает права доступа к очереди сообщений.
semget	SEM_Create	Создает набор семафоров.
semop	SEM_Op	Увеличивает или уменьшает один или несколько семафоров.
semctl	SEM_Delete	Удаляет набор семафоров.
	SEM_Owner	Изменяет принадлежность набора семафоров и права доступа к нему.
	SEM_Mode	Запрашивает права доступа к набору семафоров.
shmget	SHM_Create	Создает новый сегмент общей памяти.
shmat	SHM_Open	Вызывает процедуру **shmat** с помощью параметра **Open**.
shmat	SHM_Detach	Вызывает процедуру **shmat** с помощью параметра **Detach**.
shmctl	SHM_Close	Закрывает сегмент общей памяти.
	SHM_Owner	Изменяет принадлежность сегмента общей памяти и права доступа к нему.
	SHM_Mode	Запрашивает права доступа к сегменту общей памяти.
tcpip user level	TCPIP_config	Регистрирует в протоколе изменения интерфейса TCP/IP.
	TCPIP_host_id	Регистрирует в протоколе попытки изменить имя хоста системы.
	TCPIP_route	Регистрирует в протоколе изменения таблицы маршрутизации.
	TCPIP_connect	Вызывает процедуру **connect**.
	TCPIP_data_out	Данные отправлены.
	TCPIP_data_in	Данные получены.
	TCPIP_set_time	Регистрирует в протоколе попытки изменить системное время по сети.
tcpip kernel level	TCP_ksocket	Вызывает службы ядра TCP/IP.
	TCP_ksocketpair	
	TCP_kclose	
	TCP_ksetopt	
	TCP_kbind	
	TCP_klisten	
	TCP_kconnect	
	TCP_kaccept	
	TCP_kshutdown	
	TCP_ksend	
	TCP_kreceive	

Таблица 11. События контроля (продолжение)

Вызов пользователя или системный вызов	Событие контроля	Описание
tsm	USER_Login	Обеспечивает вход пользователя в систему.
	PORT_Locked	Указывает, что порт блокирован вследствие недопустимых попыток входа в систему.
	TERM_Logout	Обеспечивает выход пользователя из системы.
rlogind или telnetd	USER_Exit	Указывает, что пользователь вышел из системы.
usrck	USER_Check	Проверяет точность определения пользователя.
	USRCK_Error	
logout	USER_Logout	Останавливает все процессы на порте.
chsec	PORT_Change	Указывает на изменение значений атрибутов порта.
chuser	USER_Change	Изменяет атрибуты пользователя.
rmuser	USER_Remove	Удаляет пользователя.
mkuser	USER_Create	Создает пользователя.
setgroups	USER_SetGroups	Задает дополнительный ИД группы текущего процесса.
setsenv	USER_SetEnv	Задает переменную среды.
su	USER_SU	Изменяет ИД пользователя, связанный с сеансом.
grpck	GROUP_User	Удаляет несуществующих пользователей из группы.
	GROUP_Adms	Удаляет несуществующих администраторов из группы.
chgroup	GROUP_Change	Изменяет атрибуты группы.
mkgroup	GROUP_Create	Создает группу.
rmgroup	GROUP_Remove	Удаляет группу.
passwd	PASSWORD_Change	Изменяет пароль пользователя.
pwdadm	PASSWORD_Flags	Изменяет пароль администратора.
pwdck	PASSWORD_Check	Проверяет точность локальной идентификационной информации.
	PASWORD_Ckerr	
startsrc	SRC_Start	Запускает контроллер ресурсов системы.
stopsrc	SRC_Stop	Останавливает контроллер ресурсов системы.
addssys	SRC_Addssys	Добавляет определение `SRCsubsys` к объектному классу подсистем.
chssys	SRC_Chssys	Изменяет определение подсистемы в объектном классе подсистем.
addserver	SRC_Addserver	Добавляет определение субсервера к объектному классу субсерверов.
chserver	SRC_Chserver	Изменяет определение субсерверы в объектном классе субсерверов.
rmsys	SRC_Delssys	Удаляет определение подсистемы из объектного класса подсистем.
rmserver	SRC_Delserver	Удаляет определение субсервера из объектного класса `Subserver`.
enq	ENQUE_admin	Помещает файл в очередь.
qdaemon	ENQUE_exec	Планирует задания в очереди.

Таблица 11. События контроля (продолжение)

Вызов пользователя или системный вызов	Событие контроля	Описание
sendmail	SENDMAIL_Config	Управляет пересылкой почты в локальной и внешней сетях.
	SENDMAIL_ToFile	
at	AT_JobAdd	Удаляет или добавляет команды, запуск которых запланирован с помощью команды **at**.
	At_JobRemove	
cron	CRON_JobRemove	Удаляет или добавляет команды, запуск которых запланирован с помощью команды **cron**.
	CRON_JobAdd	
	CRON_Start	Указывает на запуск задания **cron**.
	CRON_Finish	Указывает на завершение задания **cron**.
nvload	NVRAM_Config	Задает доступ к памяти NVRAM.
cfgmgr	DEV_Configure	Настраивает устройства.
chdev и mkdev	DEV_Change	Указывает на изменение устройства.
mkdev	DEV_Create	Указывает, что устройство создано.
	DEV_Start	Указывает, что устройство запущено.
installp	INSTALLP_Inst	Устанавливает доступные программные продукты в совместимом установочном пакете.
	INSTALLP_Exec	
rmdev	DEV_Stop	Указывает, что устройство остановлено.
	DEV_Unconfigure	Указывает, что устройство не настроено.
	DEV_Remove	Указывает, что устройство удалено.
lchangelv, lextendlv и lreducelv	LVM_ChangeLV	Указывает, что логический том был изменен.
lchangepv, ldeletepv и linstallpv	LVM_ChangeVG	Указывает, что группа томов была изменена.
lcreatelv	LVM_CreateLV	Указывает, что логический том был добавлен в систему.
lcreatevg	LVM_CreateVG	Указывает, что группа томов была создана в системе.
ldeletepv	LVM_DeleteVG	Указывает, что группа томов была удалена из системы.
rmlv	LVM_DeleteLV	Указывает, что логический том был удален из системы.
lvaryoffvg	LVM_VaryoffVG	Деактивирует группу томов.
lvaryonvg	LVM_VaryonVG	Активирует группу томов.
Операции над логическим томом	LVM_AddLV	Добавляет логический том в существующую группу томов.
	LVM_KDeleteLV	Удаляет логический том из существующей группы томов.
	LVM_ExtendLV	Увеличивает размер логического тома путем добавления незанятых физических разделов из группы томов.
	LVM_ReduceLV	Уменьшает размер логического тома.
	LVM_KChangeLV	Изменяет существующий логический том.
	LVM_AvoidLV	Запрещает Не позволяет тому выполнять отдельные операции.

Таблица 11. События контроля (продолжение)

Вызов пользователя или системный вызов	Событие контроля	Описание
Операции над физическим томом	LVM_MissingPV	Добавляет отсутствующий физический том в существующую группу томов.
	LVM_AddPV	Добавляет физический том в существующую группу томов.
	LVM_AddMissPV	Добавляет отсутствующий физический том в существующую группу томов.
	LVM_DeletePV	Удаляет физический том из существующей группы томов.
	LVM_RemovePV	Удаляет физический том из существующей группы томов.
	LVM_AddVGSA	Добавляет область состояния группы томов (VGSA) в существующий физический том.
	LVM_DeleteVGSA	Удаляет VGSA из существующего физического тома.
Операции над группой томов	LVM_SetupVG	Настраивает группу томов путем определения логических томов и указания информации о VGSA и кэше согласования зеркальной записи (MWCC).
	LVM_DefineVG	Настраивает группу томов в ядре.
	LVM_KDeleteVG	Удаляет группу томов из ядра.
Операции резервного копирования и восстановления	BACKUP_Export	Получает состояние операции резервного копирования.
	RESTORE_Import	Получает состояние операции восстановления.
shell	USER_Shell	Получает информацию о `tty` пользователя.
reboot	USER_Reboot	Получает событие перезагрузки системы.
	PROC_Reboot	Получает событие перезагрузки процесса. Процедура **reboot** перезапускает систему или повторяет операцию загрузки начальной программы (IPL) в системе.

Настройка контроля

В этом разделе описана настройка подсистемы контроля. За дополнительной информацией обратитесь к описанию файлов конфигурации, упоминаемых в данном разделе.

1. Выберите отслеживаемые системные операции (события) из списка в файле `/etc/security/audit/events`. Если вы добавили в приложения или расширения ядра собственные события контроля, то необходимо добавить их в этот файл.
   - Добавлять в файл событие следует в том случае, если в прикладную программу включена процедура его регистрации (с помощью функции **auditwrite** или **auditlog** subroutine), либо эта программа включена в расширение ядра (с помощью служб ядра **audit_svcstart**, **audit_svcbcopy** или **audit_svcfinis**).
   - Убедитесь, что в файле `/etc/security/audit/events` есть инструкции форматирования для всех возможных типов контрольных событий. Эти инструкции необходимы для того, чтобы команда **auditpr** могла одновременно с форматированием контрольных записей выполнять запись контрольного следа.
2. Разбейте выбранные вами события на *классы контроля* (группы сходных событий). Определите классы контроля в разделе classes файла `/etc/security/audit/config`.
3. Присвойте каждому пользователю контрольные классы, а каждому контролируемому файлу (объекту) - контрольные события:

- Для того чтобы задать контрольный класс для конкретного пользователя, добавьте соответствующую строку в пользовательский раздел файла /etc/security/audit/config. Для присвоения классов контроля пользователям предназначена команда **chuser**.
- Для того чтобы связать с объектом (файлом данных или исполняемым файлом) контрольные события, укажите этот файл в соответствующем разделе файла /etc/security/audit/objects.
- Вы также можете непосредственно указать классы контроля по умолчанию для новых пользователей в файле /usr/lib/security/mkuser.default. В этом файле хранятся атрибуты пользователя, применяемые при создании новых ИД пользователей. Например, для использования класса контроля general для всех новых ИД пользователей следует указать:

  ```
  user:
      auditclasses = general
      pgrp = staff
      groups = staff
      shell = /usr/bin/ksh
      home = /home/$USER
  ```

  Для получения всех событий контроля укажите класс ALL. В этом случае даже в не очень загруженной системе будет создаваться огромный объем данных контроля. Обычно устанавливается ограничение на число записываемых событий.

4. В файле /etc/security/audit/config укажите способ сбора данных: режим лотка, режим потока или оба режима. Для данных контроля необходимо создать отдельную файловую систему, чтобы они не конкурировали за дисковое пространство с другими данными. Таким образом обеспечивается наличие достаточного дискового пространства для данных контроля. Для настройки способа сбора данных выполните следующие действия:
   - Для настройки сбора данных в режиме лотка выполните следующие действия:
     a. Включите режим лотка, указав параметр binmode = on в разделе start.
     b. Измените раздел binmode, настроив файлы-приемники и контрольный журнал, а также указав путь к файлу, содержащему базовые команды поддержки режима лотка. По умолчанию базовые команды располагаются в файле /etc/security/audit/bincmds.
     c. Убедитесь, что файлы-приемники данных контроля достаточно велики, а также задайте параметр *freespace*, указывающий, что система должна предупреждать о заполнении файловой системы.
     d. Добавьте в контрольный конвейер файла /etc/security/audit/bincmds команды оболочки, обслуживающие режим лотка.
   - Для настройки сбора данных в режиме потока выполните следующие действия:
     a. Включите режим потока, указав параметр streammode = on в разделе start.
     b. Измените раздел режима потока, указав путь к файлу с командами поддержки этого режима. По умолчанию эти команды находятся в файле /etc/security/audit/streamcmds.
     c. Добавьте в контрольный конвейер файла /etc/security/audit/streamcmds команды оболочки, обрабатывающие потоковые записи.
5. После внесения всех изменений в файлы настройки вы можете включить подсистему контроля командой **audit start**. Будет создано событие **AUD_It** со значением 1.
6. Для просмотра списка контролируемых событий и объектов воспользуйтесь командой **audit query**. Будет создано событие **AUD_It** со значением 2.
7. Для выключения подсистемы контроля воспользуйтесь командой **audit shutdown**. Будет создано событие **AUD_It** со значением 4.

Создание общего протокола контроля:

Ниже приведено несколько примеров общего протокола контроля.

В этом примере предполагается, что системный администратор хочет воспользоваться подсистемой контроля для мониторинга работы большого многопользовательского сервера. Средства интеграции с IDS не применяются, просмотр контрольных записей с целью обнаружения несанкционированных операций

выполняется вручную. Для того чтобы избежать неоправданного увеличения объема сохраняемых данных, записываются лишь сведения о некоторых основных событиях.

Решено сохранять и анализировать следующие события:

FILE_Write
Необходимо собирать информацию об операциях записи в файлы конфигурации, поэтому данное событие будет применяться для всех файлов из каталога `/etc`.

PROC_SetUserIDs
Изменения ИД пользователей

AUD_Bin_Def
Контроль конфигурации

USER_SU
Команда **su**

PASSWORD_Change
Команда **passwd**

AUD_Lost_Rec
Уведомление об утерянных записях

CRON_JobAdd
новые задания cron

AT_JobAdd
новые задания at

USER_Login
Все события входа в систему

PORT_Locked
Все блокировки терминалов из-за превышения числа неудачных попыток

Ниже приведен пример создания общего протокола контроля:

1. Выберите критические важные файлы, изменения которых необходимо отслеживать, например, все файлы в подкаталоге `/etc`, и перечислите эти файлы в событии **FILE_Write** в файле `objects`:
   ```
   find /etc -type f | awk '{printf("%s:\n\tw = FILE_Write\n\n",$1)}' >> /etc/security/audit/objects
   ```
2. С помощью команды **auditcat** настройте контроль в режиме лотка. Файл `/etc/security/audit/bincmds` выглядит примерно следующим образом:
   ```
   /usr/sbin/auditcat -p -o $trail $bin
   ```
3. Измените файл `/etc/security/audit/config`, добавив в него класс, соответствующий интересующим нас событиям. Перечислите всех существующих пользователей и укажите для них класс `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. Добавьте контрольный класс custom в файл /usr/lib/security/mkuser.default, чтобы с вновь создаваемыми ИД пользователей автоматически связывались необходимые вызовы контроля:

```
user:
    auditclasses = custom
    pgrp = staff
    groups = staff
    shell = /usr/bin/ksh
    home = /home/$USER
```

5. С помощью SMIT или команды **crfs** создайте новую файловую систему с именем /audit. Она должна быть достаточно большой для размещения двух приемных лотков и контрольного журнала необходимого размера.
6. Введите команду **audit start** и проверьте файл /audit. Первоначально вы должны обнаружить два файла приемных лотков и пустой файл trail. После того, как система поработает в течение некоторого времени, в файл trail будут добавлены контрольные записи, которые можно будет просмотреть с помощью команды:

```
auditpr  -hhelpPRtTc -v | more
```

В данном примере обрабатывается лишь несколько событий. Для просмотра всех событий необходимо указать класс ALL для всех пользователей. При этом будет создаваться огромный объем данных. Вы также можете добавить в класс custom события, относящиеся к изменению пользователей и прав доступа.

Мониторинг доступа к критически важным файлам в реальном времени:

Для отслеживания доступа к критически важным файлам в реальном времени можно воспользоваться описанным ниже способом.

Выполните следующие действия:

1. Выберите критически важные файлы, изменение которых необходимо отслеживать, например, все файлы в подкаталоге /etc, и перечислите эти файлы в событии **FILE_Write** в файле objects:

```
find /etc -type f | awk '{printf("%s:\n\tw = FILE_Write\n\n",$1)}' >> /etc/security/audit/objects
```

2. Настройте потоковый режим контроля для слежения за всеми операциями записи в файлы. (В этом примере сведения о всех операциях записи выводятся на консоль, однако в реальной среде вы можете использовать какие-либо служебные программы, которые передают события системе обнаружения несанкционированного доступа.) Файл /etc/security/audit/streamcmds выглядит примерно следующим образом:

```
/usr/sbin/auditstream | /usr/sbin/auditselect -e "event == FILE_Write" |
auditpr  -hhelpPRtTc -v > /dev/console &
```

3. Укажите в файле /etc/security/audit/config потоковый режим и добавьте класс для событий записи в файлы, а также настройке список пользователей, которых следует контролировать:

```
start:
        binmode = off
        streammode = on

stream:
        cmds = /etc/security/audit/streamcmds

classes:
        filemon = FILE_write

users:
        root = filemon
        afx = filemon
        ...
```

4. Теперь запустите команду **audit start**. На консоли будут показаны все события **FILE_Write**.

Выбор контрольных событий:

Контроль выполняется с целью обнаружения действий, угрожающих безопасности системы.

Ниже приведен список действий, нарушающих защиту системы (в случае их выполнения пользователем, не имеющим необходимых прав доступа), и поэтому подлежащих контролю:

- Действия, влияющие на работу защищенной компьютерной базы
- Идентификация пользователей
- Вход в систему
- Изменение конфигурации системы
- Попытки обойти систему контроля
- Инициализация системы
- Установка программ
- Изменение учетных файлов пользователей
- Передача данных в систему или из нее

В системе контроля не определен набор контрольных событий по умолчанию. Выбор событий или классов событий осуществляется в каждой ситуации в соответствии с конкретными требованиями.

Для того чтобы начать контроль какого-либо действия, необходимо выяснить, какая программа или процесс инициализируют контрольное событие, и указать это событие в файле `/etc/security/audit/events` вашей системы. Затем его нужно добавить либо к соответствующему классу в файле `/etc/security/audit/config`, либо в раздел объектов файла `/etc/security/audit/objects`. Список контрольных событий и инструкции по форматированию журнала приведены в файле `/etc/security/audit/events`. Сведения о способе записи и применении форматов контрольного события содержатся в описании команды **auditpr**.

После выбора событий, подлежащих контролю, следует объединить похожие события в контрольные классы. Затем контрольные классы присваиваются конкретным пользователям.

Выбор контрольных классов

Процесс присвоения контрольных событий пользователям можно упростить, объединив похожие события в контрольные классы. Их определения находятся в разделе классов файла `/etc/security/audit/config`.

Ниже приведены типичные примеры контрольных классов:

general (общие)
: Такие события изменяют состояние системы и данные идентификации пользователей. Необходимо отслеживать попытки обойти систему контроля доступа к системе.

objects (объекты)
: Обращения на запись в файлы конфигурации защиты.

kernel (ядро)
: События класса kernel формируются в ядре функциями управления процессами.

Ниже приведен пример раздела файла `/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
```

Выбор метода сбора контрольных данных

Выбор способа сбора контрольных данных зависит от того, как вы планируете использовать их в дальнейшем. Если необходимо длительно хранить большой объем данных, то выберите режим лотка. Если требуется немедленная обработка полученных данных, то рекомендуется потоковый режим. Если же требуется одновременно и возможность длительного хранения, и немедленная обработка, то включите оба режима. Ниже приведено описание каждого метода:

Режим лотка
Позволяет долго хранить контрольный след с большим объемом данных. Контрольные записи заносятся в файл, служащий временным приемным лотком. После его заполнения программа-демон **auditbin** обрабатывает данные и записывает их в контрольный след для дальнейшего хранения, а подсистема контроля в это время продолжает запись данных в другой файл.

Режим потока
Позволяет обрабатывать данные по мере их получения. Контрольные записи добавляются в циклический буфер, находящийся в ядре, и считываются из него с помощью псевдоустройства `/dev/audit`. Затем эти записи можно просмотреть на экране, напечатать или преобразовать в записи режима лотка с помощью команды **auditcat**.

Контроль за разделами рабочей схемы

В среде WPAR существует три вида контроля: глобальный, системный и контроль со стороны глобальной системы.

Контроль в глобальном WPAR и контроль внутри WPAR могут проводиться одновременно. Настройка контроля в системном WPAR и глобальном WPAR аналогична настройке в средах, отличных от WPAR. Контроль может осуществляться на уровне глобального WPAR или на уровне WPAR отдельного приложения.

Примечание: Контроль WPAR на уровне приложения нельзя включить из самого WPAR. Для этого следует использовать контроль глобального WPAR.

Контроль глобального WPAR помогает системным администраторам следить за WPAR из глобальной системы. Администратор глобальной системы управляет уровнем контроля из одного места, указывая классы контроля для каждого WPAR в глобальном файле `/etc/security/audit/config`.

В разделе "WPARS" файла `/etc/security/audit/config` администратор глобальной системы может перечислить классы контроля для отдельного WPAR. Например:

```
WPARS:
<wpar_name> = <auditclass>, ... <auditclass>
```

В предыдущем примере <имя-WPAR> должно быть именем WPAR системы, а каждый параметр "класс-контроля" должен быть определен в разделе "classes".

Чтобы настроить контроль за WPAR "testwpar", используя классы "general", "tcpip" и "lvm"; необходимо добавить в файл `/etc/security/audit/config` следующее:

```
WPARS:
testwpar = general,tcpip,lvm
```

Контроль за WPAR включается и выключается командой **audit** (она доступна администратору глобальной системы):

```
audit start -@ <имя-WPAR-1> -@ <имя-WPAR-2> ...
audit shutdown -@ <имя-WPAR-1> -@ <имя-WPAR-2> ...
```

Можно контролировать объекты WPAR из глобальной среды, указав абсолютные пути к этим объектам. Например, для контроля событий, относящихся к файлу `/wpars/wpar1/etc/security/passwd`, добавьте следующий раздел в файл `/etc/security/audit/objects` в системе AIX, в которой расположен WPAR:

```
/wpars/wpar1/etc/security/passwd:
     r = "WPAR1_PASSWD_RD"
     w = "WPAR1_PASSWD_WR"
```

Этот приведенный выше раздел анализируется в начале контроля (-@ <wpar1>), позволяя производить контроль объекта /etc/security/passwd wpar1. Эти атрибуты генерируют контрольное событие WPAR1_PASSWD_RD при каждом чтении файла /wpars/wpar1/etc/security/passwd. Эти атрибуты также генерируют контрольное событие WPAR1_PASSWD_WR при каждом открытии файла для записи.

Примечание: Необходимо включить контроль для глобальной среды перед включением контроля WPAR из глобальной среды.

Для создания отчета об осуществляемом контроле используется команда **auditpr**. Например:

```
auditpr -v < /audit/trail
```

Контроль в среде NFS

Подсистема контроля AIX поддерживает контроль смонтированных файловых систем. Конфигурация смонтированной на клиенте файловой системы подобна локальной файловой системе. Операции контроля над контролируемыми смонтированными объектами подобны локальным объектам, описанным в обзоре Контроля. Поведение контроля на клиенте и сервере для смонтированных файловых систем описано далее в этом разделе.

Контроль на клиенте NFS

Все операции, производимые клиентом над контролируемыми объектами в смонтированных файловых системах, вносятся в протокол на клиенте. Это осуществляется при условии, что сервер NFS и другие клиенты NFS не выполняют операций над этими объектами, либо на клиенте включен контроль полных путей.

Дополнительная информация приведена на странице описания команды **audit**. Если контроль полных путей не включен и файл изменяется сервером или другими клиентами, то последующий контроль может оказаться непредсказуем. Это поведение исправляется перезапуском контроля на клиенте. Если файловая система смонтирована на нескольких клиентах, то рекомендуется осуществлять контроль операций на сервере, чтобы получить точный протокол событий, либо включить контроль полных путей на клиенте.

Примечание: Конфигурациа подсистемы контроля не поддерживает использование файловой системы протоколов контроля в качестве смонтированной файловой системы NFS.

Контроль на сервере NFS

Все операции, производимые в смонтированной файловой системе как клиентом, так и сервером, вносятся в протокол на сервере NFS.

Ограничения на стороне сервера

- Если операции, выполняемые клиентом NFS не отправляются на сервер, или из-за кэширования NFS, или из-за особенностей архитектуры NFS, эти операции не будут контролироваться сервером.

  Например: После монтирования файловой системы только первая операция чтения, выполняемая для файла, контролируется сервером. Последующие операции чтения не вносятся в протокол на сервере. Это относится к операциям чтения в файлах, ссылках и каталогах.
- Операции клиента на сервере выполняются как демон **nfsd** от имени пользователя **root**.

Пример

Файловая система по имени *File_System* смонтирована на клиенте с помощью команды **mount server:/File_system /mnt**. Если файл по имени *A* в файловой системе *File_System* необходимо контролировать на сервере, то в файлах конфигурации контроля требуется настроить /File_system/A.

В вы хотите контролировать файл *A* в файловой системе *File_System* на клиенте, то необходимо настроить `/mnt/A`.

Если файл *A* настроен для контроля как на сервере, так и на клиенте, то операции, выполняемые над файлом *A* на сервере и на клиенте, контролируются и вносятся в протокол на сервере, а операции, выполняемые на клиенте, вносятся в протокол на клиенте.

Все операции, выполняемые клиентом над файлом *A*, вносятся в протокол на сервере как демон **nfsd**, а имя команды или операции.

Упрощенный протокол доступа к каталогам

Протокол LDAP определяет стандартный способ доступа к данным каталога (базы данных) и их обновления с локального или удаленного компьютера при работе в режиме клиент-сервер.

Протокол оптимизирован для чтения, просмотра и поиска в каталогах и был изначально разработан в качестве простого внешнего интерфейса к протоколу доступа к каталогу X.500. Метод доступа LDAP используется кластером для обеспечения централизованной идентификации пользователей и организации доступа к данным о пользователях и группах. Это необходимо для того, чтобы во всем кластере использовались одни и те же данные о пользователях и группах и одна и та же идентификационная информация.

Объекты LDAP хранятся в иерархической структуре, известной как дерево информации каталога (DIT). Для получения хорошего каталога необходимо начать с разработки структуры DIT. Уделите особое внимание разработке DIT, если LDAP планируется использовать для идентификации.

Загружаемый модуль идентификации LDAP

Взаимодействие LDAP с подсистемой защиты осуществляется через загрузочный модуль идентификации. Этот модуль схож с аналогичными загрузочными модулями NIS, DCE и KRB5 5. Определения загрузочных модулей хранятся в файле `/usr/lib/security/methods.cfg`.

Модуль LDAP предоставляет функции идентификации пользователей и централизованного управления пользователями и группами, основанные на применении протокола LDAP. Идентификацию можно настроить таким образом, что пользователь, определенный на сервере LDAP, сможет войти в систему клиента LDAP даже в том случае, если он не определен на этом клиенте.

Загружаемый модуль идентификации LDAP AIX полностью интегрируется в операционную систему AIX. Модуль идентификации LDAP предоставляет данные о пользователях и группах для API высокого уровня, команд и средств управления системой. Это происходит в режиме, прозрачном для пользователя. В большинстве команд высокого уровня предусмотрен флаг **-R**, предназначенный для применения различных загрузочных модулей. Например, для создания пользователя LDAP с идентификатором *joe* с системы-клиента нужно выполнить следующую команду:

```
mkuser -R LDAP joe
```

Примечание: Хотя число пользователей в группах, входящих в состав инфраструктуры LDAP, не ограничено, в процессе тестирования число пользователей, создаваемых в отдельных группах, не превышало 25000. Некоторые интерфейсы POSIX могут возвращать неполную информацию о группе. Дополнительные сведения об ограничениях такого рода приведены в документации по отдельным API.

Идентификация LDAP:

В этом разделе описаны различные ограничения, связанные с идентификацией LDAP в системе AIX.

Обратите внимание, что инфраструктура LDAP не накладывает ограничения на содержимое базы данных. Ограничения, рассмотренные в этом разделе, получены в ходе тестирования различных конфигураций. Для идентификации LDAP в AIX тестировались следующие ограничения:

Общее количество пользователей: В отдельной системе создавалось до полумиллиона пользователей. Идентификация одновременно выполнялась для нескольких сотен пользователей.

Общее число групп: В ходе тестирования в отдельной системе создавалось до 500 групп.

Максимальное количество пользователей в отдельной группе: В процессе тестирования число пользователей, создаваемых в отдельных группах, не превышало 25000.

Некоторые интерфейсы POSIX могут возвращать неполную информацию о группе. Дополнительные сведения об ограничениях такого рода приведены в документации по отдельным API. Кроме того, следует отменить, что указанные значения основаны на выполненном тестировании. Они не отрицают возможность создания в системах большего числа пользователей и групп при наличии соответствующих ресурсов.

Настройка сервера идентификационной информации IBM Tivoli Directory Server:

Для настройки системы в качестве сервера идентификационной информации LDAP, предоставляющего средства идентификации и получения данных о пользователях и группах по протоколу LDAP, нужно установить в системе пакеты клиента и сервера LDAP.

Если требуется использовать Secure Sockets Layer (SSL), то также необходимо установить пакет **GSKitV7** для IBM Tivoli Directory Server версии 6.2 (или ниже) или GSKitV8 для IBM Tivoli Directory Server версии 6.3 или выше. Администратор системы в этом случае должен создать ключ с помощью команды GSKit. Это команда **gsk7ikm** в GSKitV7 или команда **ikeyman** в GSKitV8. Дополнительная информация о настройке сервера для использования протокола SSL приведена в разделе Защита соединений с помощью SSL.

Запустите команду **mksecldap** для настройки сервера. Команда **mksecldap** задает сервер LDAP и его базу данных с именем *ldapdb2*, вводит на сервере LDAP информацию о пользователях и группах локального хоста и задает DN (отличительное имя) и пароль администратора сервера LDAP. Кроме того, эта команда может настроить протокол SSL для связи между клиентом и сервером. Команда **mksecldap** также добавляет в файл `/etc/inittab` команду запуска сервера LDAP при каждой загрузке системы.

Пользователи и группы AIX, хранятся на сервере LDAP с использованием одной из следующих схем:

Схема AIX
: Применяются классы объектов `aixAccount` и `aixAccessGroup`. Эта схема поддерживает все атрибуты пользователей и групп AIX.

Схема RFC 2307
: Включает классы объектов `posixAccount`, `shadowAccount` и `posixGroup` и используется каталогами нескольких вендоров. В схеме RFC 2307 можно сохранить только небольшое подмножество атрибутов, применяемых в AIX.

Схема RFC2307AIX
: Применяются классы объектов `posixAccount`, `shadowAccount` и `posixGroup`, `aixAuxAccount`, `aixAuxGroup`. Классы объектов `aixAuxAccount` и `aixAuxGroup` применяются для хранения атрибутов, которые используются AIX, но не поддерживаются схемой RFC 2307.

Для пользователей и групп настоятельно рекомендуется применять схему RFC2307AIX. Схема RFC2037AIX полностью совместима с RFC 2307; ее расширенные атрибуты поддерживают дополнительные функции управления пользователями AIX. Сервер IBM Tivoli Directory Server с конфигурацией схемы RFC2307AIX поддерживает не только клиенты LDAP AIX, но и другие клиенты LDAP UNIX и Linux, совместимые с RFC 2307.

Данные обо всех пользователях и группах хранятся в общем дереве AIX (с общим суффиксом). Суффикс по умолчанию равен "cn=aixdata". В команде **mksecldap** можно указать другой суффикс с помощью флага **-d**. Имена поддеревьев, создаваемых для пользователя, группы, ИД и так далее, задаются в файле конфигурации `sectoldif.cfg`. Дополнительная информация приведена в файле `sectoldif.cfg`.

Для защиты дерева AIX применяются списки управления доступом (ACL). По умолчанию ACL предоставляет права администратора только тому субъекту, который был задан в качестве администратора в параметре команды **-a**. Дополнительные права доступа предоставляются субъекту proxy, если заданы опции **-x** и **-X**. Если заданы эти опции, то создается субъект proxy с правами доступа, указанными в файле `/etc/security/ldap/proxy.ldif.template`. Создание субъекта proxy позволяет разрешить клиентам LDAP подключаться к серверу, не указывая идентификатор администратора, и таким образом ограничить права администратора клиента на сервере LDAP.

Можно запустить команду **mksecldap** на сервере LDAP, настроенному для других целей, например для поиска информации об ИД пользователей. В этом примере команда **mksecldap** добавляет дерево AIX на имеющийся сервер LDAP и вводит сведения о защите AIX. Это дерево будет защищено с помощью ACL независимо от других имеющихся деревьев.

Примечание: Следует создать резервную копию сервера LDAP перед выполнением команды **mksecldap** и расширения сервера до сервера идентификационной информации AIX.

После успешной настройки сервера идентификационной информации LDAP можно настроить тот же хост как клиент для управления пользователями и группами LDAP и предоставить пользователям LDAP право доступа для входа на этот сервер.

Если настройка сервера идентификационной информации LDAP завершается неудачно, то ее результаты можно аннулировать с помощью команды **mksecldap** с флагом **-U**. Эта операция восстановит первоначальное состояние файла `ibmslapd.conf` (или `slapd.conf`, или `slapd32.conf`). После любой неудачной попытки настройки нужно сначала выполнить команду **mksecldap** с флагом **-U**, и только после этого пытаться вновь настроить сервер с помощью команды **mksecldap**. В противном случае может сложиться ситуация, когда восстановление исходного файла конфигурации будет невозможно. По соображениям надежности в ходе операции отмены неудачной настройки не вносятся никакие изменения в базу данных, поскольку база данных могла существовать до выполнения команды **mksecldap**. Если неудавшаяся команда **mksecldap** создала базу данных, рекомендуется удалить ее вручную. Если неудавшаяся команда **mksecldap** добавила данные в существовавшую базу данных, эти данные следует удалить из базы вручную.

Понятия, связанные с данным:

Защита соединений с помощью SSL
В зависимости от способа идентификации, который используется в соединении между клиентом и сервером LDAP, пароли передаются либо в зашифрованном (unix_auth), либо в незашифрованном виде (ldap_auth). Даже в тех случаях, когда пароли передаются в зашифрованном виде по внутренней сети или по сети Internet, рекомендуется использовать протокол SSL, чтобы повысить надежность защиты. В AIX предусмотрены пакеты программ для поддержки SSL, позволяющие установить защищенное соединение между сервером каталогов и клиентом.

Информация, связанная с данной:

Команда **mksecldap**

Настройка клиента LDAP:

Для настройки клиента таким образом, чтобы он применял протокол LDAP для идентификации и получения информации о пользователях и группах, убедитесь, что на клиенте установлен пакет клиента LDAP. Если для связи с сервером применяется протокол SSL, то дополнительно требуется установить продукт GSKit, создать ключ и добавить к нему сертификат ключа сервера LDAP.

Настройка клиента с помощью команды **mksecldap** выполняется так же, как и настройка сервера LDAP. Для того чтобы клиент мог обращаться к серверу идентификационной информации LDAP, при настройке нужно указать имя сервера. Кроме того, необходимо задать DN и пароль для связывания с сервером, чтобы клиент мог получить доступ к дереву AIX на сервере. Команда **mksecldap** сохраняет DN и пароль для связывания с сервером, имя сервера, DN дерева AIX на сервере, путь к ключу SSL, пароль SSL и другие атрибуты конфигурации в файле `/etc/security/ldap/ldap.cfg`.

Команда **mksecldap** сохраняет пароль для связывания и пароль SSL (если настроен SSL) в файле `/etc/security/ldap/ldap.cfg` в зашифрованном виде. Зашифрованные пароли зависимы от системы, т.е. могут применяться только демоном **secldapclntd** той системы, в которой они создавались. Демон **secldapcIntd** может применять пароль из файла `/etc/security/ldap/ldap.cfg` как в виде чистого текста, так и зашифрованный.

Команда **mksecldap** может настроить клиент для работы с несколькими серверами. В этом случае клиент будет обращаться к серверам в том порядке, в котором они указаны, и будет работать с первым сервером, в которым удастся установить соединение. При сбое соединения между клиентом и сервером оно будет восстанавливаться по той же схеме. Сервер идентификационной информации LDAP не поддерживает переадресацию. Следует помнить о том, что серверы-копии должны быть синхронизированы с главным сервером.

Клиент подключается к серверу идентификационной информации LDAP с помощью демона клиента (**secldapcIntd**). Если на клиенте задействован загрузочный модуль LDAP, команды высокого уровня смогут обращаться к демону через библиотечные API для пользователей, определенных в LDAP. Этот демон управляет кэшем запрошенных записей LDAP. Если запрошенные данные отсутствуют в кэше, демон запрашивает сервер, обновляет кэш и возвращает информацию инициатору запроса.

В команде **mksecldap** предусмотрен ряд опций по тонкой настройке клиента - например, число нитей демона клиента, размер записи кэша и срок хранения данных в кэше. Эти опции рассчитаны на опытных пользователей. В большинстве случаев вполне удовлетворительны значения, предусмотренные по умолчанию.

На заключительном этапе настройки клиента команда **mksecldap** запускает клиентский демон и добавляет команду его запуска в файл `/etc/inittab`. Критерием успешной настройки является наличие процесса демона **secldapcIntd** в выводе команды **ls-secldapclntd**. Если сервер идентификационной информации LDAP настроен и работает, то наличие этого демона будет свидетельствовать об успешной настройке клиента.

Сервер следует настраивать в первую очередь. Настройка клиента зависит от перенесенных данных, находящихся на сервере. Для настройки клиента выполните следующие действия:

1. Установите набор файлов клиента IBM Tivoli Directory Server в операционной системе AIX.
   - В IBM Tivoli Directory Server 5.2 установите набор файлов `ldap.client`.
   - В IBM Tivoli Directory Server 7.1 установите набор файлов `idsldap`.
2. Для настройки клиента LDAP введите команду:

   ```
   # mksecldap -c -h server1.ibm.com -a cn=DN-администратора -p пароль-администратора -d cn=базовое-DN
   ```

   Укажите значения согласно вашей среде.

Информация, связанная с данной:

Команда mksecldap

Команда secldapclntd

Включение клиента для сетевых групп LDAP:

Сетевые группы (netgroups) можно применять в составе NIS-LDAP (способ обработки имени).

Для активизации клиента для сетевых групп LDAP выполните следующие действия:

1. Установите и настройте управление группами пользователей на основе LDAP, как описано в разделе “Настройка клиента LDAP” на стр. 158.

   Если установку сетевых групп выполнить не удалось, то все пользователи, определенные на сервере LDAP, будут показаны системой. Например, если *nguser* - это пользователь сетевой группы *mygroup*, определенный на сервере LDAP, то он будет показан в команде `lsuser -R LDAP nguser`.

2. Для активизации функции netgroup в определение модуля LDAP в файле /usr/lib/security/methods.cfg следует добавить атрибут options со значением netgroup. Откройте файл /usr/lib/security/methods.cfg и добавьте в раздел LDAP строку options = netgroup. Таким образом вы разрешите для загружаемого модуля LDAP поддержку сетевых групп. Например:

   ```
   LDAP:
         program = /usr/lib/security/LDAP
         program_64 =/usr/lib/security/LDAP64
         options = netgroup
   ```

   Теперь команды lsuser -R LDAP nguser, lsuser nguser и lsuser -R LDAP -a ALL не будут показывать пользователей. LDAP для этого клиента теперь рассматривается как база данных, содержащая только сетевые группы, но ни одна группа для доступа к этому клиенту еще не подключена.
3. Откройте файл /etc/passwd и добавьте строку для сетевой группы, которой необходимо предоставить доступ в систему. Например, если нужные пользователи содержатся в сетевой группе *mygroup* на сервере LDAP, то строка будет следующей:

   ```
   +@mygroup
   ```
4. Откройте файл /etc/group и добавьте строку +:, чтобы включить для групп поиск с помощью NIS:

   ```
   +:
   ```

   Теперь команда lsuser nguser вернет пользователя, поскольку *nguser* входит в сетевую группу *mygroup*.

   Команда lsuser -R LDAP nguser не найдет пользователя, а команда lsuser -R compat nguser найдет, поскольку пользователь теперь считается пользователем **compat**.
5. Для того чтобы пользователи сетевой группы могли идентифицироваться в системе, механизм идентификации AIX должен знать, какой метод применить. Если раздел значений по умолчанию в файле /etc/security/user включает строку SYSTEM = compat, то смогут идентифицироваться все пользователи сетевой группы, добавленные в файл/etc/passwd для группы. Другой вариант - настройка индивидуальных пользователей путем добавления их разделов в файл /etc/security/user вручную. Пример раздела для пользователя *nguser*:

   ```
   nguser:
           SYSTEM = compat
           registry = compat
   ```

   Теперь пользователи разрешенных сетевых групп могут идентифицироваться в системе.

   При включении сетевых групп также появляются следующие факторы:

   - Пользователи, определенные в файле /etc/security/user как члены реестра LDAP (строкиregistry=LDAP и SYSTEM="LDAP") не смогут идентифицироваться как пользователи LDAP. Теперь эти пользователи являются пользователями **nis_ldap** и для них необходимо настроить членство в стандартной сетевой группе NIS.
   - Реестр compat расширяется благодаря добавлению модулей, использующих сетевую группу. Например, если модуль LDAP поддерживает сетевую группу, то compat будет включать файлы и реестры NIS и LDAP. Значение реестра для пользователей в этих модулях будет равным compat.

Связанная информация

- Документация экспорт файлов для NFS
- Документация формат файла .rhosts для TCP/IP
- Документация формат файла hosts.equiv для TCP/IP

Поддерживаемые серверы LDAP:

Управление пользователями и группами на основе LDAP AIX поддерживает серверы IBM Tivoli Directory Server, серверы, отличные от IBM, со схемой, совместимой с RFC 2307, и серверы каталогов Microsoft Active Directory.

IBM Tivoli Directory Server

Настоятельно рекомендуется настроить управление пользователями и группами AIX с помощью серверов IBM Tivoli Directory Server. Дополнительные сведения о настройке IBM Tivoli Directory Server для управления пользователями и группами приведены в разделе Настройка сервера идентификационной информации IBM Tivoli Directory Server.

Серверы, отличные от IBM Directory Server

AIX поддерживает различные серверы каталогов, в которых пользователи и группы определяются согласно схеме RFC 2307. Будучи настроенным как клиент LDAP для таких серверов, AIX применяет серверы таким же образом, как и IBM Tivoli Directory Server со схемой RFC 2037. Эти серверы должны поддерживать протокол LDAP версии 3.

Поскольку схема RFC 2307 определяет лишь подмножество атрибутов пользователей и групп, которые может применять AIX, некоторые функции управления пользователями и группами AIX будут недоступны, если AIX настроен на использование такого сервера LDAP (например, сброс паролей пользователей, хронология паролей, ограничение на ресурсы отдельного пользователя, управление входом в некоторые системы через атрибуты `hostsallowedlogin` и `hostsdeniedlogin` AIX, производительность и т.д.).

AIX не поддерживает серверы каталогов, несовместимые с RFC 2307. Однако AIX можно настроить для работы с серверами, несовместимыми с RFC 2307, если их пользователи и группы определяются со всеми необходимыми атрибутами UNIX. Минимальный набор атрибутов пользователей и групп, необходимый AIX, - это набор, определенный в RFC 2307. Поддержка таких серверов каталогов требует настройки вручную. Для этой цели AIX предоставляет механизм отображения схемы. Дополнительная информация о формате и применении файлов схемы приведена в разделе LDAP Attribute Mapping File Format.

Microsoft Active Directory

AIX поддерживает Microsoft Active Directory (AD) в качестве сервера LDAP, предназначенного для управления пользователями и группами. На сервере AD должна быть установлена схема поддержки UNIX. Схема поддержки UNIX в AD поставляется в пакете Microsoft Service For UNIX (SFU). Определения схемы пользователей и групп несколько модифицируются в каждой очередной схеме SFU. AIX поддерживает AD в Windows 2000 и 2003 со схемой SFU версий 3.0 и 3.5, а также AD в Windows 2003 R2 с его встроенной схемой UNIX.

Из-за различий в управлении пользователями и группами в системах UNIX и системах Windows не все команды AIX применимы при работе с пользователями LDAP в случае сервера AD. Например, не работают команды **mkuser** и **mkgroup**. Однако большинство команд управления пользователями и группами работают - конечно, в зависимости от прав доступа, предоставленных идентификатору, с которым AIX подсоединяется к AD. Например, это команды **lsuser**, **chuser**, **rmuser**, **lsgroup**, **chgroup**, **rmgroup**, **id**, **groups**, **passwd** и **chpasswd**.

AIX поддерживает два механизма идентификации пользователей для серверов Windows: LDAP и Kerberos. В обоих случаях AIX поддерживает идентификацию пользователей по протоколу LDAP для AD, не требуя наличия соответствующей учетной записи пользователя в AIX.

Настройка операционной системы AIX для работы с Active Directory с использованием LDAP:

AIX поддерживает Microsoft Active Directory (AD) в качестве сервера LDAP, предназначенного для управления пользователями и группами. Необходимо, чтобы на сервере AD была установлена схема поддержки UNIX.

Администратор может использовать команду **mksecldap** для настройки AIX на сервере AD тем же образом, что и на сервере IBM Tivoli Directory Server. Для упрощения процесса команда **mksecldap** скрывает все подробности настройки. Перед запуском команды **mksecldap** для настройки AIX на сервере AD:

1. На сервере AD должна быть установлена схема поддержки UNIX.
2. На сервере AD должны существовать пользователи, для которых поддерживается UNIX.

Дополнительные сведения об установке схемы UNIX на AD и предоставления пользователям AD поддержки UNIX приведены в соответствующей документации Microsoft.

Схема AD часто имеет множественные определения атрибутов для одного и того же атрибута UNIX (например, существуют множественные определения пароля пользователя и члена группы). Хотя AIX поддерживает большинство из них, при выборе используемых определений необходимы тщательное рассмотрение и планирование. Во избежание конфликтов рекомендуется использовать общий AD и общее определение в системах AIX и других системах помимо AIX.

Выбор атрибута пароля активного каталога:

AIX поддерживает два механизма идентификации: **unix_auth** и **ldap_auth**.

Для использования механизма **unix_auth** необходимо, чтобы пароль активного каталога Microsoft Active Directory (AD) был в зашифрованном формате. При идентификации зашифрованный пароль будет получен из AD и сравнен с зашифрованным представлением пароля, введенного пользователем. Если они совпадают, то идентификация успешна. В режиме **ldap_auth** AIX идентифицирует пользователя с помощью операции связывания с сервером LDAP, передавая пользовательский профайл и пароль. Пользователь идентифицирован, если операция связывания завершена успешно. AD поддерживает множественные атрибуты пароля пользователя. Различные режимы идентификации AIX требуют указания различных атрибутов пароля пользователя AD.

режим unix_auth

В режиме **unix_auth** можно использовать следующие атрибуты пароля AD:

- **userPassword**
- **unixUserPassword**
- **msSFU30Password**

Управление паролями в AIX может быть затруднено из-за множественности атрибутов пароля AD. Запоминая, какие атрибуты управления паролями будут использоваться клиентами UNIX, можно запутаться. Возможность преобразования атрибутов LDAP в системе AIX позволяет настроить управление паролями требуемым образом.

По умолчанию при работе AD в Windows 2000 и 2003 AIX использует атрибут **msSFU30Password**, а в Windows 2003 R2 - **userPassword**. Для использования другого пароля потребуется изменить файл `/etc/security/ldap/sfu30user.map` (или файл `/etc/security/ldap/sfur2user.map` при работе AD в Windows 2003 R2). Найдите строку, которая начинается со слова **spassword**, и замените значение в третьем поле строки на имя требуемого атрибута пароля AD. Дополнительная информация приведена в разделе Формат файла преобразования атрибута LDAP. Для настройки клиента LDAP AIX после изменения выполните команду **mksecldap**. Если клиент LDAP AIX уже настроен, то выполните команду **restart-secldapclntd** для перезапуска демона **secldapclntd** и применения изменений.

В режиме **unix_auth** пароль может быть не синхронизирован между Windows и UNIX, что приведет к наличию различных паролей в разных системах. Это происходит при смене пароля с AIX на Windows, поскольку Windows использует атрибут пароля **uncodepwd**. Команда AIX **passwd** может сбросить пароль UNIX, в результате чего он станет тем же, что и в Windows, но AIX не поддерживает автоматической смены пароля Windows при изменении пароля UNIX из AIX.

режим ldap_auth

В Active Directory также существует атрибут пароля **unicodepwd**. Этот атрибут пароля используется системами Windows для идентификации пользователей Windows. В операции связывания с AD следует указывать пароль **unicodePwd**. Ни один пароль, указанный в режиме **unix_auth**, не сработает при операции связывания. Если в командной строке указать опцию **ldap_auth**, то команда **mksecldap** преобразует атрибут пароля в атрибут AD **unicodePwd** в настройках клиента без необходимости во внесении изменений вручную.

Преобразуя пароли AIX с атрибутом **unicodePwd**, пользователи, определенные в AD, могут входить в системы Windows и AIX с одним и тем же паролем. Пароль, сброшенный из системы AIX или Windows, остается действительным как для AIX, так и для Windows.

Выбор атрибута члена группы активного каталога:

Служба Microsoft для UNIX определяет атрибуты члена группы **memberUid**, **msSFU30MemberUid** и **msSFU30PosixMember**.

Атрибуты **memberUid** и **msSFU30MemeberUid** в качестве значений принимают имена учетных записей пользователей, а **msSFU30PosixMember** - только полное DN. Например, для учетной записи пользователя *foo* (с фамилией *bar*), определенной в AD:

- **memberUid: foo**
- **msSFU30MemberUid: foo**
- **msSFU30PosixMember: CN=foo bar,CN=Users,DC=austin,DC=ibm,DC=com**

Операционная система AIX поддерживает все эти атрибуты. Чтобы определить, какой атрибут следует использовать, обратитесь к администратору AD. По умолчанию команда **mksecldap** настраивает операционную систему AIX для использования атрибута **msSFU30PosixMember** для AD, выполняющегося в Windows 2000 и 2003, а атрибут **uidMember** - для AD, выполняющегося в Windows 2003 R2. Различие обусловлено тем, что AD выбирает эти атрибуты при добавлении пользователя в группу в системе Windows. Для поддержки различных платформ деловая стратегия может потребовать изменения атрибута члена группы, который установлен по умолчанию.

Если атрибут члена группы должен быть другим, то преобразование можно изменить путем редактирования файла преобразования группы. Файлом преобразования группы для AD является `/etc/security/ldap/sfu30group.map`, выполняемый в системах Windows 2000 и 2003, и файл `/etc/security/ldap/sfur2group.map` - в Windows 2003 R2. Найдите строку, которая начинается со слова **users**, и замените значение в третьем поле именем требуемого атрибута членов группы. Дополнительная информация приведена в разделе Формат файла преобразования атрибута LDAP. Для настройки клиента LDAP в системе AIX после внесения изменений выполните команду **mksecldap**, а если система AIX уже настроена, то выполните команду **restart-secldapclntd** для перезапуска демона **secldapclntd** и применения изменений.

Множественные подразделения организации:

На вашем сервере AD может быть определено несколько подразделений организации, каждое из которых содержит набор пользователей.

Большинство пользователей Windows AD определены в поддереве **cn=users,...**, но некоторые могут быть определены вне этого поддерева. Для такого сервера AD можно использовать поддержку множественных базовых DN, реализованную в AIX. Дополнительная информация приведена в разделе Поддержка множественных базовых DN.

Идентификация Kerberos для серверов Windows:

Кроме механизмов идентификации LDAP операционная система AIX поддерживает также идентификацию пользователей по протоколу Kerberos для серверов Windows.

AIX поддерживает идентификацию Kerberos для идентификации Windows KDC и LDAP для Windows Active Directory путем создания составного загрузочного модуля KRB5ALDAP. Поскольку идентификационная информация пользователя извлекается из Microsoft Active Directory, нет необходимости создавать в AIX соответствующие учетные записи пользователя.

Работа с пользователями LDAP:

Пользователями и группами на сервере идентификационной информации LDAP можно управлять с любого клиента LDAP с помощью команд высокого уровня.

В большинстве команд высокого уровня предусмотрен флаг **-R**, позволяющий задать альтернативный загрузочный модуль идентификации, например LDAP, DCE, NIS или KRB5 5. Дополнительную информацию о флаге **-R** можно найти в описании конкретной команды.

Для того чтобы разрешить пользователю идентификацию через сервер LDAP, присвойте атрибуту **SYSTEM** этого пользователя значение LDAP с помощью команды **chuser**. Атрибуту **SYSTEM** можно присвоить несколько значений, и тогда пользователь сможет проходить идентификацию через разные загрузочные модули (например, compat и LDAP). Дополнительная информация о методах идентификации пользователей приведена в разделе “Идентификация пользователей” на стр. 71 и в описании формата атрибута **SYSTEM** файла `/etc/security/user`.

Пользователей можно перевести на идентификацию через LDAP при настройке клиента, если указать в команде **mksecldap** флаг **-u**:

1. Введите команду:

   ```
   mksecldap -c -u пользователь1,пользователь2,...
   ```

   где *пользователь,пользователь2,...* - список пользователей. В этом списке можно указывать как локальных пользователей системы, так и удаленных пользователей LDAP. Атрибуту **SYSTEM** указанных пользователей в файле `/etc/security/user` будет присвоено значение LDAP. После этого данные пользователи смогут входить в систему только через LDAP. Указанные пользователи должны быть определены на сервере идентификационной информации LDAP, так как в противном случае они не смогут входить в систему с данного хоста. С помощью команды **chuser** можно присвоить атрибуту **SYSTEM** комбинированное значение, чтобы пользователь мог пользоваться разными методами идентификации (например, local и LDAP).
2. Введите

   ```
   mksecldap -c -u ALL
   ```

   Эта команда присваивает атрибуту **SYSTEM** всех локальных пользователей в файле `/etc/security/user` значение LDAP. Все эти пользователи смогут входить в систему только через LDAP. Локальные пользователи должны быть определены на сервере идентификационной информации LDAP, так как в противном случае они не смогут входить в систему с данного хоста. Пользователи, определенные на сервере LDAP, но не определенные в локальной системе, также не смогут входить в систему с данного хоста. Для того чтобы разрешить удаленным пользователям LDAP входить в систему с данного хоста, присвойте атрибуту **SYSTEM** этих пользователей значение LDAP с помощью команды **chuser**.

Кроме того, можно разрешить вход в систему любым пользователям LDAP, независимо от того, определены ли они в локальной системе. Для этого нужно указать в разделе "default" файла `/etc/security/user` значение "LDAP" для атрибута SYSTEM. Для всех пользователей, для которых атрибут **SYSTEM** не задан явно, применяется атрибут из раздела "default". Например, если в разделе default указан атрибут `"SYSTEM = "compat""`, то, изменив его на `"SYSTEM = "compat OR LDAP""`, вы разрешите идентификацию пользователей как средствами AIX, так и через LDAP. Если в разделе default будет указан атрибут`"SYSTEM = "LDAP""`, то такие пользователи смогут входить в систему только через LDAP. На пользователей, для которых явно задан атрибут **SYSTEM**, значение этого атрибута из раздела default не распространяется.

Поддержка множественных базовых DN:

AIX поддерживает несколько базовых DN. В файле `/etc/security/ldap/ldap.cfg` можно указать до 10 базовых DN для каждого элемента.

Приоритет базовых DN соответствует порядку их следования в файле `/etc/security/ldap/ldap.cfg`. В случае со множественными базовыми DN команды AIX выполняют операции в соответствии с приоритетом базовых DN таким образом:

- Операция запроса (например, в случае вызова команды **lsuser**) выполняется над базовыми DN в соответствии с их приоритетом, пока не будет найдена совпадающая учетная запись. Если же не совпало ни одно базовое DN, будет возвращено сообщение об ошибке. При запросе ALL будут возвращены все учетные записи каждого базового DN.
- Операция изменения (например, в случае вызова команды **chuser**), производится с первой совпадающей учетной записью.
- Операция удаления (например, в случае вызова команды **rmuser**), производится с первой совпадающей учетной записью.
- Операция создания (например, в случае вызова команды **mkuser**), производится только с первым базовым DN. AIX не поддерживает создание учетных записей для других базовых DN.

Администратор сервера несет ответственность за то, чтобы в базе данных учетных записей отсутствовали конфликты. Если для одной и той же учетной записи существует несколько определений в различных поддеревьях, то видимым для AIX будет только первый из них. Операция поиска возвращает только первую совпадающую учетную запись. Подобным образом операции изменения и удаления производятся только с первой совпавшей учетной записью.

При использовании для настройки клиента LDAP команда **mksecldap** найдет базовые DN для каждого элемента и сохранит их в файле `/etc/security/ldap/ldap.cfg`. Когда на сервере для одного элемента существует несколько базовых DN, команда **mksecldap** случайным образом выбирает одно из них. Для работы AIX с несколькими базовыми DN необходимо изменить файл `/etc/security/ldap/ldap.cfg` после успешного выполнения команды **mksecldap**. Найдите соответствующее базовое DN и добавьте требуемое дополнительное базовое DN. AIX поддерживает до 10 базовых DN для каждого элемента. Остальные базовые DN игнорируются.

AIX поддерживает также пользовательские фильтры и область поиска для каждого базового DN. Базовое DN может иметь собственный фильтр и собственную область поиска, которые у других DN могут быть иными. Фильтры можно использовать для определения набора учетных записей, которые видны AIX.

AIX видны только те учетные записи, которые удовлетворяют условиям фильтра.

Настройка SSL на сервере LDAP:

Для настройки Secure Sockets Layer (SSL) на сервере LDAP установите наборы файлов LDAP и **GSKit** для включения поддержки шифрования на сервере. Эти наборы файлов можно найти на компакт-диске пакета расширенияAIX.

Для включения поддержки SSL при идентификации с помощью сервера каталоговIBM выполните следующие действия.

1. Установите IBM Tivoli Directory Server **GSKit** для IBM Tivoli Directory Server версии 6.2 или **GSKitv8** для IBM Tivoli Directory Server версии 6.3, если он еще не установлен.
2. Создайте личный ключ и сертификат сервера IBM Directory Server с помощью соответствующей утилиты управления ключами GSKit. Используйте утилиту **gsk7ikm** с IBM Tivoli Directory Server версии 6.2, а инструмент ikeyman - для IBM Tivoli Directory Server версии 6.3 или выше. Сертификат сервера может быть подписан коммерческой сертификационной компанией (CA), например, VeriSign, либо самой утилитой GSKit. Скопируйте в базу данных ключей клиентских приложений сертификат CA (либо собственный сертификат).

3. Сохраните файл базы данных ключей сервера и связанный файл хранения паролей на сервере. По умолчанию база данных ключей расположена в каталоге /usr/ldap/etc.
4. Запустите следующую команду для настройки сервера, где **mykey.kdb** - база данных ключей, а *keypwd* - пароль базы данных ключей:

   ```
   # mksecldap -s -a cn=admin -p pwd -S rfc2307aix -k /usr/ldap/etc/mykey.kdb -w keypwd
   ```

Настройка SSL на клиенте LDAP:

Для работы с SSL на клиенте LDAP установите наборы файлов ldap.max_crypto_client и GSKit с компакт-диска пакета расширения AIX.

Для включения поддержки SSL для LDAP после настройки сервера выполните следующие действия:

1. Запустите gsk7ikm для создания базы данных ключей на каждом клиенте.
2. Скопируйте сертификат сервера на каждый из клиентов. Если SSL сервера использует собственный сертификат, этот сертификат следует экспортировать в первую очередь.
3. Импортируйте сертификат сервера в базу данных ключей на каждом клиенте с помощью команды gsk7ikm.
4. Включите SSL на каждом клиенте:

   ```
   # mksecldap -c -h servername -a adminDN -p pwd -k /usr/ldap/etc/mykey.kdb -p keypwd
   ```

   Здесь /usr/ldap/etc/mykey.kdb - это полный путь к базе данных ключей, а *keypwd* - пароль ключа. Если в командной строке не введен пароль ключа, будет использован сохраненный файл паролей из этого же каталога. Имя этого файла должно совпадать с именем базы данных ключей, а расширение должно быть **.sth** (например, mykey.sth).

Управление доступом к хосту с помощью LDAP:

В AIX управление доступом к системе организовано на уровне пользователей. Конфигурацию пользователей LDAP можно задать таким образом, что они смогут входить в систему AIX только через LDAP. Для этого атрибуту **SYSTEM** этих пользователей присваивается значение.

Атрибут **SYSTEM** находится в файле /etc/security/user. Для изменения его значения можно воспользоваться командой **chuser**:

```
# chuser -R LDAP SYSTEM=LDAP registry=LDAP foo
```

Примечание: Не следует указывать значение LDAP для атрибута **SYSTEM** в разделе default, если только вы не хотите разрешить всем пользователям LDAP входить в систему.

Эта команда разрешает пользователю *foo* вход в данную систему. Кроме того, она присваивает атрибуту registry значение LDAP, что позволяет вести протокол попыток пользователя *foo* войти в систему и выполнять иные операции над пользователем на сервере LDAP.

Если администратору требуется разрешить пользователю LDAP вход в систему на различных клиентах, эту операцию нужно выполнить на каждом из этих клиентов.

Пользователям LDAP можно разрешить вход в систему AIX только с конкретных систем-клиентов LDAP. Эта возможность позволяет осуществлять централизованное управление доступом к хостам. Администратор может создать два списка управления доступом для каждого пользователя: список разрешений и список запретов. Оба списка будут храниться в учетной записи пользователя на сервере LDAP. Пользователю будет разрешен доступ к системам и сетям, указанным в списке разрешений, и запрещен доступ к системам и сетям, указанным в списке запретов. Если система указана и в списке разрешений, и в списке запретов, доступ к ней будет запрещен. Предусмотрено два способа создания списков управления доступом: при создании пользователя (команда **mkuser**) и при его изменении (с помощью команды **chuser**).

В целях обратной совместимости в случае, если для пользователя не задан ни список разрешений, ни список запретов, ему будет разрешен вход во все системы-клиенты LDAP.

Ниже приведено несколько примеров работы со списками управления доступом:

```
# mkuser -R LDAP hostsallowedlogin=host1,host2 foo
```

Эта команда создает пользователя *foo*, при этом *foo* может входить в систему только на хостах `host1` и `host2`.

```
# mkuser -R LDAP hostsdeniedlogin=host2 foo
```

Эта команда создает пользователя *foo*, при этом пользователю *foo* разрешен вход в систему на любых хостах, за исключением `host2`.

```
# chuser -R LDAP hostsallowedlogin=192.9.200.1 foo
```

Эта команда изменяет пользователя *foo* таким образом, что он может входить в систему только с клиента с адресом `192.9.200.1`.

```
# chuser -R LDAP hostsallowedlogin=192.9.200/24 hostsdeniedlogin=192.9.200.1 foo
```

Эта команда изменяет пользователя *foo* таким образом, что ему разрешено входить в систему с любых хостов из подсети `192.9.200/24`, за исключением хоста `192.9.200.1`.

Дополнительная информация приведена в описании команды **chuser**.

Защита соединений с помощью SSL:

В зависимости от способа идентификации, который используется в соединении между клиентом и сервером LDAP, пароли передаются либо в зашифрованном (unix_auth), либо в незашифрованном виде (ldap_auth). Даже в тех случаях, когда пароли передаются в зашифрованном виде по внутренней сети или по сети Internet, рекомендуется использовать протокол SSL, чтобы повысить надежность защиты. В AIX предусмотрены пакеты программ для поддержки SSL, позволяющие установить защищенное соединение между сервером каталогов и клиентом.

Дополнительная информация приведена в разделах:

- “Настройка SSL на сервере LDAP” на стр. 165
- “Настройка SSL на клиенте LDAP” на стр. 166

Использование режима только аутентификации LDAPA:

Модуль LDAP является полнофункциональным модулем, который поддерживает как аутентификацию, так и идентификацию пользователя. Модуль LDAPA предоставляет режим только аутентификации. Модуль LDAPA подобен модулю LDAP, но позволяет использовать режим только аутентификации.

В режиме только аутентификации модуль LDAPA должен быть скомбинирован с другим модулем базы данных для формирования составного модуля, а не автономного модуля. Модуль LDAPA выполняет аутентификацию пользователя, в то время как второй модуль выполняет идентификацию. Этот комбинированный модуль называется составным модулем. Необходимо определить пользователей как на сервере LDAP, так и на сервере базы данных для этого составного модуля.

С помощью модуля LDAPA информация о группе получается из сервера базы данных. Например, в случае файлов LDAPA информация о группе получается из локального файла `/etc/group`. Если некоторые из пользователей LDAP принадлежат только группам LDAP, необходимо создать соответствующие группы LDAP на сервере базы данных перед настройкой модуля файлов LDAPA. Создав соответствующую группу, можно избежать ситуацию, в которой пользователь файлов LDAPA не может обработать параметр группы, потому что он не существует на сервере базы данных.

Примечание: Модуль LDAPA не поддерживает создание и удаление пользователей. Для того чтобы создать пользователя файлов LDAPA, системный администратор должен создать пользователя LDAP с помощью модуля LDAP, а затем создать такого же пользователя локально. Затем необходимо сделать его пользователем файлов LDAPA, установив для атрибута SYSTEM пользователя и реестра значение LDAPAfiles с помощью команды **chuser**.

Для того чтобы настроить LDAP в режиме только аутентификации с помощью модуля LDAPA, выполните команду **mksecldap** с опцией -i <*модуль_базы_данных*>. Эта команда создает модуль LDAPA с options = authonly и составной модуль загрузки LDAPA <*модуль_базы_данных*>.

Например, для настройки LDAP в режиме только аутентификации и использования локальных файлов для модуля базы данных используйте следующий пример:

```
mksecldap -c -h <сервер ldap>
-a <dn связывания> -p
<пароль связывания> -i
файлы
```

В файл /usr/lib/security/methods.cfg добавляется следующее:

```
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
```

В разделе LDAPA параметр options = authonly указывает на установку модуля LDAPA в режим только аутентификации. Раздел LDAPAfiles определяет составной модуль загрузки.

Модуль LDAP сохраняется для обработки данных, отличных от пользователя и группы, например, RBAC. Модуль LDAP по-прежнему может использоваться как автономный модуль аутентификации, независимый от модуля LDAPA.

Информация, связанная с данной:

Команда mksecldap

Поддерживаемые атрибуты LDAPA:

Модуль LDAPA в режиме только идентификации поддерживает ограниченное количество атрибутов стратегии пароля AIX. Остальные атрибуты AIX задаются модулем базы данных.

Модуль LDAPA только аутентификации поддерживает следующие атрибуты:

- maxage
- minage
- minlen
- lastupdate
- флаги
- 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

Не все серверы LDAP поддерживают эти атрибуты. Когда сервер LDAP поддерживает не все перечисленные атрибуты, поддерживаются только те атрибуты, которые являются общими как для этого списка, так и для файла преобразования пользовательских атрибутов. Файл преобразования находится в каталоге `/etc/security/ldap`.

Для сервера, совместимого с RFC2307, без поддержки схемы AIX поддерживаются следующие атрибуты AIX:

- maxage
- minage
- lastupdate
- pwdwarntime
- lastupdate

Связывание Kerberos:

Кроме простого связывания, предусматривающего применение DN и пароля связывания, **secldapclntd** поддерживает связывание с помощью разрешений Kerberos V.

Для применения связывания Kerberos необходимо предоставить демону **secldapclntd** доступ к ключам субъекта связывания из файла keytab. Если применяется связывание Kerberos, демон **secldapclntd** выполняет идентификацию Kerberos на сервере LDAP с помощью имени субъекта и файла keytab, указанных в файле конфигурации клиента `/etc/security/ldap/ldap.cfg`. Кроме того, в этом случае демон **secldapclntd** игнорирует DN и пароль связывания, указанные в файле `/etc/security/ldap/ldap.cfg`.

После успешной идентификации Kerberos демон **secldapclntd** сохраняет удостоверения связывания в каталоге `/etc/security/ldap/krb5cc_secldapclntd`. Эти удостоверения используются для последующего связывания. Через час после создания разрешений для повторного связывания с сервером LDAP потребуется повторная инициализация демона **secldapclntd** с продлением разрешений.

Настройка клиента LDAP для применения связывания Kerberos выполняется с помощью команды **mksecldap**, в которой должны быть указаны DN и пароль связывания. После успешной настройки добавьте в файл `/etc/security/ldap/ldap.cfg` правильные значения связанных атрибутов Kerberos. Связывание Kerberos применяется в ходе перезапуска демона **secldapclntd**. После успешной настройки отличительное имя (DN) связывания и пароль связывания больше не используются. Их можно безопасно удалить или закомментировать в файле `/etc/security/ldap/ldap.cfg`.

Создание субъекта Kerberos:

Для поддержки связывания Kerberos необходимо создать в Центре рассылки ключей (KDC) по крайней мере два субъекта для сервера и клиента IDS. Первый из них - это субъект сервера LDAP, а второй - субъект, применяемый клиентами для связывания с сервером.

Ключи субъектов следует заменить в файле keytab, поскольку с их помощью должны запускаться процесс сервера и процесс демона клиента.

Следующий пример основан на службе сетевой идентификации IBM. Если программное обеспечение Kerberos установлено из других источников, соответствующие команды могут отличаться от приведенных ниже.

- Запустите инструмент kadmin на сервере KDC от имени пользователя root.

  ```
  #/usr/krb5/sbin/kadmin.local
  kadmin.local:
  ```

- Создайте субъект ldap/*имя-хоста-сервера* для сервера LDAP. *Имя-хоста-сервера* - это полное отличительное имя хоста, в котором выполняется сервер LDAP.

  ```
  kadmin.local: addprinc ldap/plankton.austin.ibm.com
  Внимание: Не указана стратегия для "ldap/plankton.austin.ibm.com@ud3a.austin.ibm.com":
  Еще раз введите пароль для субъекта "ldap/plankton.austin.ibm.com@ud3a.austin.ibm.com":
  Субъект "ldap/plankton.austin.ibm.com@ud3a.austin.ibm.com" создан.
  kadmin.local:
  ```

- Создайте для нового субъекта сервера файл keytab. Данный ключ будет применяться сервером LDAP при запуске. Для создания файла keytab с именем `slapd_krb5.keytab` выполните следующую команду:

  ```
  kadmin.local: ktadd -k /etc/security/slapd_krb5.keytab ldap/plankton.austin.ibm.com
  Запись для субъекта ldap/plankton.austin.ibm.com, версия ключа - 2,
  тип шифрования - Triple DES, режим cbc с HMAC/sha1, добавлена в keytab
  WRFILE:/etc/security/slapd_krb5.keytab.
  Запись для субъекта ldap/plankton.austin.ibm.com, версия ключа - 2,
  тип шифрования - ArcFour с HMAC/md5, добавлена в файл keytab
  WRFILE:/etc/security/slapd_krb5.keytab.
  Запись для субъекта ldap/plankton.austin.ibm.com, версия ключа - 2,
  тип шифрования - AES-256, режим CTS с 96-разрядным SHA-1 HMAC, добавлена в
  файл keytab
  WRFILE:/etc/security/slapd_krb5.keytab.
  Запись для субъекта ldap/plankton.austin.ibm.com, версия ключа - 2,
  тип шифрования - DES, режим cbc с RSA-MD5, добавлена в файл keytab
  WRFILE:/etc/security/slapd_krb5.keytab.
  kadmin.local:
  ```

- Создайте субъект ldapadmin для администратора IDS.

  ```
  kadmin.local: addprinc ldapadmin
  Внимание: Не указана стратегия ldapadmin@ud3a.austin.ibm.com; по умолчанию стратегия не применяется.
  Обратите внимание, что стратегия может быть переопределена ограничениями ACL.
  Введите пароль для субъекта "ldapadmin@ud3a.austin.ibm.com":
  Еще раз введите пароль для субъекта "ldapadmin@ud3a.austin.ibm.com":
  Субъект "ldapadmin@ud3a.austin.ibm.com" создан.
  kadmin.local:
  ```

- Создайте для субъекта связывания `kdapadmin.keytab` файл keytab. Данный ключ может применяться демоном **secldapclntd** клиента.

  ```
  kadmin.local: ktadd -k /etc/security/ldapadmin.keytab ldapadmin
  Запись для субъекта ldapadmin, версия ключа - 2, тип шифрования -
  Triple DES, режим cbc с HMCA/sha1, добавлена в файл keytab
  WRFILE:/etc/security/ldapadmin.keytab.
  Запись для субъекта ldapadmin, версия ключа - 2, тип шифрования -
  ArcFour с HMAC/md5, добавлена в файл keytab
  WRFILE:/etc/security/ldapadmin.keytab.
  Запись для субъекта ldapadmin, версия ключа - 2, тип шифрования -
  AES-256 режим CTS с 96-разрядным SHA-1 HMAC, добавлена в файл keytab
  WRFILE:/etc/security/ldapadmin.keytab.
  ```

```
Запись для субъекта ldapadmin, версия ключа - 2, тип шифрования -
DES, режим cbc с RSA-MD5, добавлена в файл keytab
WRFILE:/etc/security/ldapadmin.keytab.
kadmin.local
```

- Создайте субъект ldapproxy, применяемый для связывания клиентов с сервером LDAP.

```
kadmin.local: addprinc ldapproxy
Внимание: Не указана стратегия ldapproxy@ud3a.austin.ibm.com; по умолчанию стратегия не применяется.
Обратите внимание, что стратегия может быть переопределена ограничениями ACL
Введите пароль для субъекта "ldapproxy@ud3a.austin.ibm.com":
Еще раз введите пароль для субъекта "ldapproxy@ud3a.austin.ibm.com":
Субъект "ldapproxy@ud3a.austin.ibm.com" создан.
kadmin.local:
```

- Создайте для субъекта связывания **ldapproxy** файл keytab с именем *ldapproxy.keytab*. Данный ключ может применяться демоном **secldapclntd** клиента.

```
kadmin.local: ktadd -k /etc/security/ldapproxy.keytab ldapproxy
Запись для субъекта ldapproxy, версия ключа - 2, тип шифрования -
Triple DES, режим cbc с HMAC/sh1, добавлена в файл keytab
WRFILE:/etc/security/ldapproxy.keytab.
Запись для субъекта ldapproxy, версия ключа - 2, тип шифрования -
ArcFour с HMAC/md5, добавлена в файл keytab
WRFILE:/etc/security/ldapproxy.keytab
Запись для субъекта ldapproxy, версия ключа - 2, тип шифрования -
AES-256, режим CTS с 96-разрядным SHA-1 HMAC, добавлена в файл keytab
WRFILE:/etc/security/ldapproxy.keytab
Запись для субъекта ldapproxy, версия ключа - 2,
тип шифрования - DES, режим cbc с RSA-MD5, добавлена в файл keytab
WRFILE:/etc/security/ldapproxy.keytab.
kadmin.local:
```

Применение связывания Kerberos для сервера IDS:

Алгоритм включения связывания Kerberos для сервера IDS.

В следующем примере показано, каким образом можно настроить сервер IDS для применения связывания Kerberos.

Тестирование этого примера было выполнено с помощью IDS v5.1:

1. Установите набор файлов `krb5.client`.
2. Убедитесь, что файл `/etc/krb5/krb5.conf` существует и настроен правильным образом. При необходимости вы можете настроить этот файл с помощью команды **/usr/sbin/config.krb5**.

```
# config.krb5 -r ud3a.austin.ibm.com -d austin.ibm.com -c KDC -s alyssa.austin.ibm.com
Инициализация конфигурации...
Создание /etc/krb5/krb5_cfg_type...
Создание /etc/krb5/krb5.conf...
Команда выполнена успешно.
# 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. Найдите файл keytab субъекта ldap:/*имя-хоста-сервера* и скопируйте его в каталог `/usr/ldap/etc`. Например: `/usr/ldap/etc/slapd_krb5.keytab`.
4. Укажите права доступа таким образом, чтобы процесс сервера мог обращаться к этому файлу.

   ```
   # chown ldap:ldap/usr/ldap/etc/slapd_krb5.keytab
   #
   ```

5. Для того чтобы настроить сервер IDS для применения связывания Kerberos, откройте файл `/etc/ibmslapd.conf` и добавьте следующую строку:

   ```
   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. Создайте запись преобразования между субъектом ldapproxy и DN связывания cn-proxyuser,cn=aixdata.
   a. Если запись DN связывания на сервере IDS уже создана, создайте файл `ldapproxy.ldif` со следующим содержимым:

      ```
      dn: cn=proxyuser,cn=aixdata
      changetype: modify
      add: objectclass
      objectclass: ibm-securityidentities
      -
      add:altsecurityidentities
      altsecurityidentities: Kerberos:ldapproxy@ud3a.austin.ibm.com
      ```

      или

   b. Если запись DN связывания на сервере IDS не создана, создайте файл `proxyuser.ldif` со следующим содержимым:

   Примечание: Вместо *proxyuserpwd* укажите свой пароль.

   ```
   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
   ```

   Добавьте созданную запись DN связывания на сервер IDS с помощью команды **ldapmodify**.

   ```
   # ldapmodify -D cn-admin -w adminPwd -f /tmp/proxyuser.ldif
   (изменяется запись cn=proxyuser,cn=mytest)
   #
   ```

7. Перезапустите сервер IDS.

Применение связывания Kerberos для клиента LDAP AIX:

Клиент LDAP AIX можно настроить для применения исходного связывания Kerberos с сервером LDAP.

Сервер IDS должен быть настроен таким образом, чтобы хост сервера был своим же клиентом.

Тестирование этого примера было выполнено с помощью IDS v 5.1:

1. Установите набор файлов krb5.client.
2. Убедитесь, что файл /etc/krb.conf существует и настроен правильным образом. При необходимости вы можете настроить этот файл с помощью команды **/usr/sbin/config.krb5**.
3. Найдите файл keytab субъекта связывания и скопируйте его в каталог /etc/security/ldap.
4. Укажите права доступа 600.
5. Настройте клиента с помощью команды **mksecldap**, в которой должны быть указаны DN и пароль связывания. Убедитесь, что команды AIX применимы к пользователям LDAP.
6. Откройте файл /etc/security/ldap/ldap.cfg и укажите в нем связанные атрибуты Kerberos. В следующем примере применяются субъект связывания **ldapproxy** и файл keytab **ldapproxy.keytab**. Для получения прав доступа администратора IDS замените *ldapproxy* на *ldapadmin*, а также *ldapproxy.keytab* на *ldapadmin.keytab*.

   ```
   useKRB5:yes
   krbprincipal:ldapproxy
   krbkeypath:/etc/security/ldap/ldapproxy.keytab
   krbcmddir:/usr/krb5/bin/
   ```

   Теперь DN и пароль связывания можно удалить из файла ldap.cfg или выделить их комментариями, поскольку демон **secldapclntd** настроен для работы со связыванием Kerberos.
7. Перезапустите демона **secldapclntd**.
8. Теперь файл /etc/security/ldap/ldap.cfg можно распространить на другие клиенты.

Контроль сервера идентификационной информации LDAP:

В продукте SecureWay Directory 3.2 и более поздних версий предусмотрен встроенный модуль ведения протокола контроля сервера. Если этот модуль будет включен, он будет заносить в протокол данные об операциях сервера LDAP. Дополнительная информация об этом модуле приведена в документации по LDAP в руководстве *Описание установочного комплекта LPP*.

Функция контроля сервера идентификационной информации, поставляемая с операционной системой AIX, называется *модуль проверки защиты LDAP*. Этот модуль работает независимо от стандартной службы контроля SecureWay Directory, поэтому его и стандартную службу можно применять независимо друг от друга. Модуль контроля AIX заносит в протокол только те события, которые связаны с обновлением или просмотром информации, относящейся к защите сервера LDAP в AIX. Этот модуль работает в рамках общей среды контроля AIX.

В файле /etc/security/audit/event создаются следующие события для LDAP:

- LDAP_Bind
- LDAP_Unbind
- LDAP_Add
- LDAP_Delete
- LDAP_Modify
- LDAP_Modifydn
- LDAP_Search

Кроме того, в файле /etc/security/audit/config создается определение класса контроля ldapserver.

Для того чтобы включить контроль сервера идентификационной информации LDAP, добавьте следующую строку в раздел каждого пользователя в файле /etc/security/audit/config:

```
ldap = ldapserver
```

Поскольку модуль контроля сервера идентификационной информации LDAP выполнен в виде компонента среды контроля AIX, он является частью подсистемы контроля AIX. Для включения и отключения модуля применяются стандартные системные команды **audit start** и **audit shutdown**. Все контрольные записи

добавляются в контрольный след, который можно просмотреть с помощью команды **auditpr**. Дополнительная информация приведена в разделе “Обзор подсистемы контроля” на стр. 136.

Команды LDAP:

В этом разделе приведены команды LDAP.

команда lsldap

Команда **lsldap** служит для отображения субъектов службы присвоения имен с настроенного сервера LDAP. Субъектами могут быть псевдонимы, функции automount, параметры загрузки, интерфейсы ether, группы, хосты, сетевые группы, сети, утилита passwd, протоколы, утилиты rpc и службы.

команда mksecldap

Команда **mksecldap** применяется для настройки средств идентификации и управления данными на серверах и клиентах IBM SecureWay Directory. Эта команда должна быть выполнена на сервере и на всех клиентах.

демон secldapclntd

Демон **secldapclntd** получает запросы от загрузочного модуля LDAP, передает их на сервер LDAP Security Information Server, обрабатывает ответ, полученный от этого сервера, и передает результаты обработки загрузочному модулю LDAP.

Команды управления LDAP:

Существует несколько команд для управления LDAP.

команда start-secldapclntd

Команда **start-secldapclntd** запускает демон **secldapclntd**, если он еще не запущен.

команда stop-secldapclntd

Команда **stop-secldapclntd** завершает работу демона **secldapclntd**.

команда restart-secldapclntd

Сценарий **restart-secldapclntd** завершает работу демона **secldapclntd**, а затем вновь запускает демон. Если демон **secldapclntd** не запущен, этот сценарий запускает его.

команда ls-secldapclntd

Команда **ls-secldapclntd** показывает сведения о состоянии демона **secldapclntd**.

команда flush-secldapclntd

Команда **flush-secldapclntd** удаляет содержимое кэша демона **secldapclntd**.

Команда sectoldif

Команда **sectoldif** считывает информацию о пользователях и группах, определенных в локальной системе, и выдает результат в поток вывода в формате **ldif**.

Формат файла преобразования для атрибутов LDAP:

Файлы атрибутов используются модулем /usr/lib/security/LDAP и демоном **secldapclntd** для преобразования имен атрибутов AIX в имена атрибутов LDAP.

Каждая запись в этом файле задает правило преобразования для одного атрибута. Запись состоит из четырех полей, разделенных пробелами:

```
AIX_Attribute_Name AIX_Attribute_Type LDAP_Attribute_Name LDAP_Value_Type
```

Ниже приведены описания этих полей:

AIX_Attribute_Name
Имя атрибута AIX.

AIX_Attribute_Type
Тип атрибута AIX. Допустимы значения SEC_CHAR, SEC_INT, SEC_LIST и SEC_BOOL.

LDAP_Attribute_Name
Имя атрибута LDAP.

LDAP_Value_Type
Тип атрибута LDAP. Допустимы значения **s** (простое значение) и **m** (комбинированное значение).

LDAP и KRB5LDAP на одном клиенте

Если LDAP находится в составе сложного модуля, такого как KRB5LDAP, то можно выполнять только операции чтения, но не записи. Однако, после внесения указанных ниже изменений в файл **/usr/lib/security/methods.cfg**, LDAP и составные модули загрузки, такие как KRB5LDAP, согласуются в одном файле. Для этого выполните следующие действия:

1. Настройте клиент LDAP и клиенты KRB5LDAP обычным образом.
2. Измените файл **/usr/lib/security/methods.cfg** следующим образом:

```
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. Измените раздел по умолчанию в файле **/etc/security/user** следующим образом:

```
        SYSTEM = "KRB5LXAP OR LDAP OR compat"
```

   Пользователи LDAP могут быть обработаны обычным образом. В следующих примерах показана обработка пользователей KRB5LDAP:

```
mkuser -R KRB5LXAP <имя_пользователя>
rmuser -R KRB5LXAP <имя_пользователя>
lsuser -R KRB5LXAP <имя_пользователя>
passwd -R KRB5LXAP <имя_пользователя>
```

Файловая система с шифрованием EFS

Файловая система с шифрованием позволяет отдельным пользователям системы шифровать свои данные в файловой системе J2 с использованием индивидуальных хранилищ ключей.

С каждым пользователем связан ключ. Эти ключи хранятся в криптографически защищенном хранилище ключей и при успешном входе ключи пользователя загружаются в ядро и связываются с идентификационными данными процессов. Впоследствии, когда процессу необходимо открыть файл, защищенный EFS, эти идентификационные данные тестируются и если найден ключ, соответствующий защите файла, то процесс может расшифровать ключ файла и, следовательно, его содержимое. Управление ключами на групповой основе также поддерживается.

Примечание: EFS является частью общей стратегии защиты. Она предназначена для работы в комплексе с процедурами и мерами контроля для надежной защиты компьютеров.

Удобство работы с файловой системой с шифрованием

В обычном режиме работы управление ключами, шифрование и расшифровка файлов в файловой системе с шифрованием (EFS) просты для пользователей.

EFS является частью базовой операционной системы AIX. Для включения EFS и создания среды EFS пользователь root (или любой пользователь с правами RBAC **aix.security.efs**, дополнительные сведения приведены в разделе Субъекты EFS) должен использовать команду **efsenable**. Это включение выполняется один раз. После включения EFS при входе пользователя в систему его ключ и хранилище ключей создаются без участия пользователя и защищаются или зашифровываются паролем пользователя для входа в систему. Впоследствии ключи пользователей автоматически используются файловой системой J2 при шифровании и расшифровке файлов EFS. Каждый файл EFS защищен собственным уникальным ключом файла, а этот ключ файла, в свою очередь, защищен или зашифрован ключом владельца файла или ключом группы в зависимости от прав доступа к файлу.

По умолчанию EFS отключена для файловой системы J2. Когда EFS включена, файловая система J2 явным образом управляет шифрованием и расшифровкой в ядре при получении запросов на чтение и запись. Команды управления пользователями и группами (например, **mkgroup**, **chuser** и **chgroup**) явным образом управляют хранилищами ключей пользователей и групп.

Для того чтобы пользователи могли управлять ключами и шифрованием файлов, в EFS предусмотрены следующие команды:

efskeymgr
Для управления ключами и их администрирования

efsmgr Управляет шифрованием файлов, каталогов и файловой системы

Субъекты файловой системы с шифрованием

Существует три типа пользователей, которые могут управлять ключами EFS и использовать их:

Полный и ограниченный доступ под управлением ИД root:

Доступ к ключам под управлением ИД root может быть неограниченным или ограниченным. В любом из этих режимов ИД root не может получить доступ к зашифрованному файлу или хранилищу ключей пользователя, просто войдя в систему под учетной записью другого пользователя с помощью **su**.

В первом режиме ИД root может сбросить пароль хранилища ключей пользователя и получить доступ к ключам пользователя в этом хранилище ключей. Этот режим обеспечивает более гибкие возможности администрирования системы.

Во втором режиме ИД root может сбросить пароль пользователя для входа в систему, но не может сбросить пароль хранилища ключей пользователя. Пользователь root не может сменить пользователя (с помощью команды **su**) и унаследовать открытое хранилище ключей. Хотя пользователь root может создавать и удалять пользователей и группы вместе со связанными с ними хранилищами ключей, он не может получить доступ к ключам, содержащимся в этих хранилищах. Этот режим обеспечивает большую степень защиты от атаки со стороны злонамеренного пользователя root.

Существуют два режима управления и использования хранилищ ключей: Root Admin и Root Guard. Также предусмотрен ключ администрирования EFS.

Ключ администрирования EFS позволяет хранить пароль ко всем хранилищам ключей в режиме Root Admin. Этот ключ расположен в особом хранилище ключей **efs_admin**. Доступ к особому хранилищу ключей **efs_admin** предоставляется только пользователя м с правами доступа (пользователю root и группе безопасности при установке или идентификации RBAC **aix.security.efs**).

Когда хранилище ключей находится в режиме Root Guard, содержащиеся в нем ключи не могут быть получены без правильного пароля хранилища ключей. Это обеспечивает надежную защиту от злонамеренного пользователя root, но может вызвать затруднения, если пользователь забывает свой пароль, поскольку невозможно восстановить пароль без потери ключей в хранилище. В результате пользователь больше не сможет получить доступ к своим данным. В этом режиме работы хранилища некоторые операции не могут выполняться немедленно, и их выполнение откладывается. Ожидающие операции появляются, например, в случаях добавления или подавления создания ключа группового доступа в хранилище ключей пользователя или восстановления личного ключа. Ими управляет владелец хранилища ключей.

Ключ администратора efs_admin:

Хранилище ключей efs_admin содержит особый ключ, который может открыть любое хранилище ключей пользователя или группы в режиме root admin (режим по умолчанию).

Пароль для открытия этого особого хранилища ключей при включении EFS хранится в пользователе root и защищенных хранилищах ключей групп. Этот пароль можно передать другим группам и пользователям или удалить с помощью команды **efskeymgr**. При совместном использовании этого ключа с идентификацией RBAC **aix.security.efs** пользователь может администрировать EFS (то есть, обращаться к хранилищам ключей в режиме root admin).

Замечания по efs_admin RBAC

В системах со включенным Управлением доступом на основе ролей команда **efs_admin** защищена идентификацией **aix.security.efs**.

Хранилище ключей пользователя:

В большинстве обычных операций управление ключами производится автоматически. Для обслуживания и расширенного использования EFS используется команда **efskeymgr**. Пользователи могут создавать шифрованные файлы и каталоги с помощью команды **efsmgr**. Управление хранилищем ключей производится при выполнении большинства команд управления пользователями. При добавлении пользователя в группу он автоматически получает доступ к хранилищу ключей группы.

Владелец файла с правами доступа EFS может использовать команду **efsmgr** для предоставления доступа EFS другим пользователям и группам (владельцы файлов имеют подобные полномочия на ACL в UNIX). Смена паролей пользователями не затрагивает отдельные процессы, выполняемые под тем же ИД пользователя с открытым хранилищем ключей.

Хранилище ключей файловой системы с шифрованием

Хранилища ключей защищены паролями. Пользователи могут выбрать пароль хранилища ключей, который отличается от пароля для входа в систему. В таком случае хранилище ключей не открывается и является недоступным при обычном входе пользователя. Пользователь должен загрузить хранилище ключей вручную с помощью команды **efskey**, чтобы ввести пароль хранилища ключей.

Формат хранилища ключей - **PKCS # 12**. Хранилища ключей расположены в следующих файлах:

хранилище ключей пользователя
/var/efs/users//keystore

хранилище ключей групп
/var/efs/groups//keystore

хранилище ключей efsadmin
/var/efs/efs_admin/keystore

Если пользователь задает один и тот же пароль для входа в систему и для хранилища ключей, то хранилище ключей открывается и активируется при входе.

Пользователь может выбрать тип алгоритма шифрования и длину ключа с помощью команды EFS **efskeymgr**.

Доступ к хранилищу ключей наследуется всеми дочерними процессами.

Управление ключами на групповой основе также поддерживается. Если хранилище ключей находится в режиме защиты, то только члены групп могут добавлять и удалять ключи групп в хранилищах ключей члена. Хранилище ключей пользователя содержит личный ключ пользователя, а также пароль для открытия хранилищ ключей групп пользователя, которые содержат личные ключи группы.

Примечание: Хранилище ключей EFS автоматически открывается как часть стандартного входа AIX только если пароль хранилища ключей пользователя совпадает с паролем для входа. Это установка по умолчанию, задаваемая при создании хранилища ключей пользователя. Способы входа, которые отличаются от стандартного входа AIX, например загружаемые модули идентификации и подключаемые модули идентификации, не могут автоматически открывать хранилище ключей.

Шифрование и наследование

EFS является функцией J2. Опция системы **efs** должна иметь значение **да** (см. команды **mkfs** и **chfs**).

J2 EFS автоматически зашифровывает и расшифровывает пользовательские данные. Тем не менее, если пользователь имеет права доступа на чтение файла EFS, но не имеет требуемого ключа, то он не сможет прочесть файл обычным способом. Если у пользователя нет действительного ключа, то расшифровать данные невозможно.

Все криптографические функции обеспечены службами CLiC и библиотеками пользователя CLiC.

По умолчанию EFS отключена для файловой системы J2. Для активации наследования EFS и шифрования данных пользователя необходимо включить EFS для файловой системы J2. Файл создается как зашифрованный либо явно с помощью команды **efsmgr** или неявно путем наследования EFS. Наследование EFS можно активировать на уровне файловой системы и/или на уровне каталога.

Команда **ls** перечисляет записи зашифрованного файла с предшествующей **e**.

Команды **cp** и **mv** могут обрабатывать метаданные и зашифрованные данные по сценариям EFS-to-EFS и EFS-to-non-EFS.

Команды **backup**, **restore** и **tar**, а также связанные с ними команды могут создавать резервные копии и восстанавливать зашифрованные данные, в том числе метаданные EFS, которые используются для шифрования и расшифровки.

Резервное копирование и восстановление

Очень важно надлежащим образом управлять архивированием или созданием резервных копий хранилищ ключей, связанных с архивированными файлами EFS. Также необходимо обеспечить пароли, связанные с хранилищами ключей, когда они заархивированы или созданы их резервные копии. Невыполнение любой из этих задач может привести к потере данных.

При создании резервных копий шифрованных файлов EFS можно использовать параметр **-Z** команды **backup**, чтобы копия файла в шифрованном виде создавалась вместе с криптографическими метаданными

файла. Данные и метаданные файла защищены надежным шифрованием. Таким образом используется преимущество защиты резервной копии файла с помощью надежного шифрования. Необходимо создавать резервную копию хранилища ключей владельца файла и группы, связанных с файлом, для которого проводится резервное копирование. Эти хранилища ключей расположены в следующих файлах:

хранилища ключей пользователей
/var/efs/users/*user_login*/*

хранилища ключей групп
/var/efs/groups//keystore

хранилище ключей efsadmin
/var/efs/efs_admin/keystore

Для восстановления резервной копии EFS используется команда **restore** (резервная EFS копия создается командой **backup** с параметром **-Z**). Команда восстановления гарантирует восстановление криптографических метаданных. Восстановление хранилищ ключей из резервной копии при восстановлении файлов не является необходимым, если пользователь не изменял ключи в личном хранилище ключей. Когда пользователь изменяет пароль хранилища ключей, внутренний ключ хранилища не изменяется. Для изменения внутренних ключей хранилища используется команда **efskeymgr**.

Если внутренний ключ хранилища ключей пользователя остается неизменным, то пользователь может сразу же открыть и расшифровать восстановленный файл с помощью текущего хранилища ключей. Тем не менее, если внутренний ключ хранилища ключей пользователя изменен, то пользователь должен открыть хранилище ключей, резервная копия которого при создании была связана с резервной копией файла. Это хранилище ключей может быть открыто с помощью команды **efskeymgr –o**. Команда **efskeymgr** запрашивает пароль пользователя для открытия хранилища ключей. Это тот пароль, который связан с хранилищем ключей во время создания резервной копии.

Предположим, что хранилище ключей пользователя Боба было защищено паролем **foo** (пароль ‘foo’ не является защищенным паролем, и используется только в данном примере для его упрощения), а резервная копия зашифрованных файлов Боба была создана в январе с хранилищем ключей Боба. В этом примере Боб также использует **foo** в качестве пароля для входа в AIX. В феврале Боб изменил пароль на **bar**, что повлекло за собой изменение пароля для доступа к его хранилищу ключей на **bar**. Если в марте файлы EFS Боба были восстановлены, то Боб сможет открыть и просмотреть эти файлы, используя свое текущее хранилище ключей и пароль, поскольку он не изменил внутренний ключ хранилища ключей.

Тем не менее, если Бобу было необходимо сменить внутренний ключ хранилища ключей (при помощи команды **efskeymgr**), то по умолчанию прежний внутренний ключ хранилища ключей будет помечен как устаревший и оставлен в хранилище ключей Боба. Когда пользователь обращается к файлу, EFS автоматически определяет, что восстановленный файл использовал прежний внутренний ключ, и для расшифровки файла она использует ключ, помеченный как устаревший. Во время того же сеанса доступа EFS преобразует файл для использования нового внутреннего ключа. Процесс не оказывает значительное воздействие на производительность системы, потому что обработка производится с помощью хранилища ключей и криптографических метаданных файла, и повторного шифрования данных файла не требуется.

Если помеченный на удаление внутренний ключ удален с помощью **efskeymgr**, то прежнее хранилище ключей, содержащее прежний внутренний ключ, должно восстанавливаться и использоваться совместно с файлами, зашифрованными с помощью этого внутреннего ключа.

По этой причине возникает вопрос о том, каким образом безопасно хранить и архивировать прежние пароли. Существуют методы и инструменты для архивирования паролей. Обычно они предусматривают создание файла со списком всех прежних паролей, а затем шифрование этого файла и его защиту с использованием текущего хранилища ключей, которое, в свою очередь, защищено текущими паролями. Тем не менее, среды ИТ и стратегии безопасности различны в разных организациях, поэтому необходимо рассматривать и обдумывать конкретные требования к защите в вашей организации для разработки стратегии безопасности и процедур, которые более всего пригодны для вашей среды.

Внутренний механизм J2 EFS

Каждый файл J2 EFS связан с особым расширенным атрибутом, который содержит метаданные EFS, используемые для проверки crypto authority и информации для шифрования и расшифровки файлов (ключей, алгоритма шифрования и т.д.).

J2 не распознает содержимое EA. Для определения crypto authority (прав доступа) каждого файла EFS требуются идентификационные данные пользователя и метаданные EFS.

Примечание: Следует уделять особое внимание ситуациям, в которых файл или данные могут быть утеряны (например, при удалении EA файла).

Наследование защиты EFS

После того, как для каталога активирована EFS, для всех создаваемых прямых дочерних объектов автоматически активируется EFS, если эта установка не переопределена вручную. Атрибуты EFS родительского каталога имеют приоритет перед атрибутами EFS файловой системы.

Область наследования каталога равно ровно одному уровню. Все создаваемые дочерние объекты наследуют атрибуты EFS родительского каталога, если для родительского каталога активирована EFS. Существующие дочерние объекты сохраняются в текущем зашифрованном или незашифрованном состоянии. Логическая цепочка наследования разрывается при смене атрибутов EFS родительского объекта. Эти изменения не распространяются на существующие дочерние объекты каталога и должны применяться для них отдельно.

Замечания о разделах рабочей схемы

Перед включением или использованием файловой систем с шифрованием для раздела рабочей схемы следует сначала включить EFS в глобальной системе с помощью команды **efsenable**. Это включение должно проводиться только один раз. Кроме того, все файловые системы, в том числе, с использованием EFS, должны создаваться в глобальной системе.

Настройка файловой системы с шифрованием

Вначале следует выполнить следующие действия.

Настройку рекомендуется проводить таким образом.

1. Установите набора файлов **clic.rte**. Этот набор файлов содержит криптографические библиотеки и расширения ядра, требуемые EFS. Набор файлов **clic.rte** находится в Пакете расширений AIX.
2. Включите EFS в системе с помощью команды **efsenable** (например, >efsenable –a). В приглашении на ввод пароля рекомендуется ввести корневой пароль. Хранилища ключей пользователей создаются автоматически, а затем пользователь входит в систему после запуска команды **efsenable**. После выполнения в системе команды **efsenable –a** EFS включена, и больше не требуется запускать команду **efsenable**.
3. С помощью опции **–a efs=yes** создайте файловую систему с активированной EFS. Например, `crfs -v jfs2 -m /foo -A yes -a efs=yes -g rootvg -a size=20000`
4. После монтирования файловой системы включите криптографическое наследование в системе с активированной EFS. Это можно сделать с помощью команды **efsmgr**. Продолжая предыдущий пример, в котором создана файловая система **/foo**, выполните следующую команду: `efsmgr -s -E /foo`. Благодаря этому каждый файл, создаваемый или используемый в этой системе, будет зашифрованным.

Впоследствии, когда пользователь или процесс с открытым хранилищем ключей будет создавать файл в этой системе, данный файл будет шифрованным. Когда пользователь или файл читает этот файл, производится автоматическая расшифровка этого файла для пользователей, обладающих правами доступа к нему.

Дополнительные сведения приведены в следующих источниках:

- Команды **chfs**, **chgroup**, **chuser**, **cp**, **efsenable**, **efskeymgr**, **efsmgr**, **lsuser**, **ls**, **mkgroup**, **mkuser** и **mv**
- Файлы `/etc/security/group` and `/etc/security/user`

Удаленный доступ к хранилищам ключей Зашифрованной файловой системы

В среде предприятия можно централизовать хранилища ключей Зашифрованной файловой системы (EFS). При сохранении баз данных, которые управляют хранилищами ключей, независимо на каждой системе, может оказаться трудно управлять хранилищами ключей. Централизованное хранилище ключей EFS в AIX позволяет хранить базы данных хранилища ключей пользователя и группы на сервере Упрощенного протокола доступа к каталогам (LDAP), что позволяет централизовано управлять хранилищем ключей EFS.

Понятия, связанные с данным:

“Упрощенный протокол доступа к каталогам” на стр. 156
Протокол LDAP определяет стандартный способ доступа к данным каталога (базы данных) и их обновления с локального или удаленного компьютера при работе в режиме клиент-сервер.

Обзор удаленного доступа к хранилищам ключей Зашифрованной файловой системы:

В этом разделе описываются базы данных Зашифрованной файловой системы (EFS), включение LDAP для команд EFS и доступ к уникальному хранилищу ключей.

Все базы данных хранилища ключей AIX EFS можно хранить в LDAP, включающем следующие базы данных EFS:

- Хранилище ключей пользователя
- Хранилище ключей группы
- Хранилище ключей администратора
- Cookie

Операционная система AIX предоставляет утилиты для помощи в выполнении следующих задач управления:

- Экспорт данных локального хранилища ключей на сервер LDAP
- Настройка клиента для использования данных хранилища ключей EFS и LDAP
- Управление доступом к данным хранилища ключей EFS
- Управление данными LDAP из системы клиента

Все команды управления базой данных хранилища ключей EFS могут использовать базу данных хранилища ключей LDAP. Если порядок поиска по всей системе не задан в файле `/etc/nscontrol.conf`, то операции хранилища ключей зависят от пользователя и атрибута efs_keystore_access группы. Если атрибут efs_keystore_access установлен как ldap, то команды EFS выполняют операции хранилища ключей в хранилище ключей LDAP.

В следующей таблице описаны изменения команд EFS для LDAP.

Таблица 12. Включение команд EFS для LDAP

Команда	**Информация LDAP**
Все команды EFS	Когда атрибут efs_keystore_access установлен как ldap, не обязательно использовать специальную опцию -L *домен* в любой команде для выполнения операций хранилища ключей в LDAP.
efskeymgr	Включает в себя опцию -L load_module, так что можно выполнять явно операции хранилища ключей в LDAP.
efsenable	Включает в себя опцию -d Basedn, так что можно выполнять начальную установку в LDAP для настройки хранилища ключей EFS. Начальная установка включает в себя добавление базовых отличительных имен (DN) для хранилища ключей EFS и создание локальной структуры каталогов (/var/efs/).

Таблица 12. Включение команд EFS для LDAP (продолжение)

Команда	Информация LDAP
efskstoldif	Генерирует данные хранилища ключей EFS для LDAP из следующих баз данных в локальной системе: • /var/efs/users/*имя_пользователя*/keystore • /var/efs/groups/*имя_группы*/keystore • /var/efs/efs_admin/keystore • Cookies, если они существуют, для всех хранилищ ключей

Все записи хранилища ключей должны быть уникальны. Каждая запись хранилища ключей соответствует DN записи, которая содержит имя группы и пользователя. Система запрашивает ИД пользователей (uidNumber), ИД групп (gidNumber) и DN. Запрос выполняется успешно, когда имена пользователя и группы совпадают с соответствующими DN. Перед созданием или миграцией записей хранилища ключей EFS в LDAP, убедитесь в том, что имена группы и пользователя и ИД в системе уникальны.

Задачи, связанные с данной:

"Экспорт данных хранилища ключей Зашифрованной файловой системы в LDAP"
Необходимо заполнить данные хранилища ключей на сервере LDAP для его использования в качестве централизованного хранилища для хранилища ключей Зашифрованной файловой системы (EFS).

"Настройка клиента LDAP для хранилища ключей Зашифрованной файловой системы"
Для использования данных хранилища ключей Зашифрованной файловой системы (EFS), хранящихся в LDAP, необходимо настроить систему в качестве клиента LDAP.

Экспорт данных хранилища ключей Зашифрованной файловой системы в LDAP:

Необходимо заполнить данные хранилища ключей на сервере LDAP для его использования в качестве централизованного хранилища для хранилища ключей Зашифрованной файловой системы (EFS).

Перед созданием или миграцией записей хранилища ключей EFS в LDAP, убедитесь в том, что имена группы и пользователя и ИД в системе уникальны.

Для того чтобы заполнить данные хранилища ключей EFS на сервере LDAP, выполните следующие действия:

1. Установите схему хранилища ключей EFS для LDAP на сервере LDAP:
   a. Загрузите схему хранилища ключей EFS для LDAP из файла `/etc/security/ldap/sec.ldif` в системе AIX.
   b. Выполните команду **ldapmodify** для обновления схемы сервера LDAP с помощью схемы хранилища ключей EFS для LDAP.
2. Выполните команду **efskstoldif** для чтения данных из локальных файлов хранилища ключей EFS и вывода данных в подходящем для LDAP формате. Для того чтобы установить доступ к уникальному хранилищу ключей, поместите данные хранилища ключей EFS, которые расположены в LDAP, под таким же родительским отличительным именем (DN), что и у данных группы и пользователя.
3. Сохраните данные в файле.
4. Выполните команду **ldapadd -b** для заполнения данных хранилища ключей на сервере LDAP.

Понятия, связанные с данным:

"Обзор удаленного доступа к хранилищам ключей Зашифрованной файловой системы" на стр. 181
В этом разделе описываются базы данных Зашифрованной файловой системы (EFS), включение LDAP для команд EFS и доступ к уникальному хранилищу ключей.

Настройка клиента LDAP для хранилища ключей Зашифрованной файловой системы:

Для использования данных хранилища ключей Зашифрованной файловой системы (EFS), хранящихся в LDAP, необходимо настроить систему в качестве клиента LDAP.

Для того чтобы настроить клиент LDAP для хранилища ключей EFS, выполните следующие действия:

1. Для настройки системы как клиента LDAP выполните команду **/usr/sbin/mksecldap**. Команда **mksecldap** динамически находит указанный сервер LDAP, чтобы определить расположение данных хранилища ключей EFS. Затем она сохраняет результаты в файле `/etc/security/ldap/ldap.cfg`. Команда **mksecldap** определяет расположение для данных хранилища ключей пользователя, группы, администратора и efscookies.
2. Выполните одно из следующих действий, чтобы включить LDAP как домен поиска данных хранилища ключей EFS:
   - Установите для атрибута **`efs_keystore_access`** пользователя и группы значение **`file`** или **`ldap`**.
   - Определите порядок поиска для хранилища ключей на уровне системы с помощью файла `/etc/nscontrol.conf`. В следующей таблице показан пример.

Таблица 13. Пример конфигурации файла `/etc/nscontrol.conf`

Атрибут	Описание	Порядок поиска (secorder)
efsusrkeystore	Этот порядок поиска является общим для всех пользователей.	LDAP, файлы
efsgrpkeystore	Этот порядок поиска является общим для всех групп.	файлы, LDAP
efsadmkeystore	Этот порядок поиска находит хранилище ключей администратора для любого целевого хранилища ключей.	LDAP, файлы

> **Внимание:** Конфигурация, определенная в файле `/etc/nscontrol.conf`, переопределяет значение, установленное для атрибута **`efs_keystore_access`** пользователя или группы. Это же верно для атрибута **`efs_adminks_access`**.

После настройки системы в качестве клиента LDAP и включения LDAP как домена поиска для данных хранилища ключей EFS, демон клиента /usr/sbin/secldapclntd извлекает данные хранилища ключей EFS из сервера LDAP при каждом выполнении какой-либо операции хранилища ключей LDAP.

Понятия, связанные с данным:

“Обзор удаленного доступа к хранилищам ключей Зашифрованной файловой системы” на стр. 181
В этом разделе описываются базы данных Зашифрованной файловой системы (EFS), включение LDAP для команд EFS и доступ к уникальному хранилищу ключей.

Стандарт шифрования с общим ключом #11

Подсистема шифрования с общим ключом (PKCS #11) предоставляет приложениям независимый от аппаратного обеспечения интерфейс для работы с аппаратными устройствами (маркерами).

Информация в этом разделе соответствует версии 2.20 стандарта PKCS #11.

Подсистема PKCS #11 использует следующие компоненты:

- Общий объект API (`/usr/lib/pkcs11/ibm_pks11.so`) - это универсальный интерфейс для драйвера устройства, в котором реализована поддержка стандарта PKCS #11. Такая многоуровневая структура позволяет пользователю применять новые устройства PKCS #11 без повторной компиляции существующих приложений.
- Драйвер устройств PKCS #11, предоставляющий приложениям возможности, аналогичные возможностям, предоставляемым другим компонентам ядра, таким как EFS или IPSec.
- Если платформа поддерживает функции шифровального сопроцессора, то драйвер устройства PKCS #11 использует аппаратное ускорение, доступное в операциях AES, SHA и HMAC. Для повышения производительности можно включить привязку сетевой памяти.

Информация, связанная с данной:

Поддержка привязки памяти в AIX

Шифровальный сопроцессор IBM 4758, модель 2

Шифровальный сопроцессор IBM 4758, модель 2 обеспечивает защиту среды обработки данных.

Перед настройкой подсистемы PKCS #11 убедитесь, что в адаптере установлен поддерживаемый микрокод.

Шифровальный ускоритель IBM 4960

Шифровальный ускоритель IBM 4960 дает возможность ускорения транзакций шифрования. Перед настройкой подсистемы PKCS #11 убедитесь в правильности настройки адаптера.

Проверка совместимости шифровального сопроцессора IBM 4758, модель 2 с подсистемой PKCS #11:

Подсистема PKCS #11 автоматически обнаруживает адаптеры, поддерживающие вызовы PKCS #11, в процессе установки и загрузки. По этой причине, с неправильно настроенными шифровальными сопроцессорами IBM 4758 модели 2 будет невозможно работать через интерфейс PKCS #11, и при отправке вызовов в адаптер будут возникать ошибки.

Для проверки конфигурации адаптера выполните следующие действия:

1. Введите следующую команду, чтобы проверить программное обеспечение адаптера:
   ```
   lsdev -Cc adapter | grep crypt
   ```

   Если в выводе команды отсутствует информация о шифровальном сопроцессоре IBM 4758, модель 2, то убедитесь в том, что карта правильно вставлена в разъем и установлено необходимое для нее программное обеспечение.
2. С помощью следующей команды, проверьте, загружено ли на карту необходимое встроенное программное обеспечение:
   ```
   csufclu /tmp/l ST дополнительный-номер-устройства
   ```

   Убедитесь в том, что в образ сегмента 3 загружено приложение PKCS #11. Если оно не загружено, обратитесь к инструкциям по получению и установке последней версии микрокода, приведенным в документации адаптера.

   Примечание: Недоступность этой утилиты означает отсутствие установленной программной поддержки адаптера.

Проверка совместимости шифровального ускорителя IBM 4960, модель 2 с подсистемой шифрования с общим ключом #11:

Подсистема PKCS #11 автоматически обнаруживает адаптеры, поддерживающие вызовы PKCS #11, в процессе установки и загрузки. По этой причине с неправильно настроенными шифровальными ускорителями IBM 4960 будет невозможно работать через интерфейс PKCS #11, а при отправке вызовов в адаптер будут возникать ошибки.

Для проверки правильности установки программного обеспечения адаптера введите следующую команду:

```
lsdev -Cc adapter | grep ica
```

Если в выводе команды отсутствует информация о шифровальном ускорителе IBM 4960, то убедитесь в том, что карта правильно вставлена в разъем и установлено необходимое для нее программное обеспечение.

Конфигурация подсистемы шифрования с общим ключом #11

Подсистема PKCS #11 автоматически обнаруживает устройства, поддерживающие PKCS #11. Однако для некоторых приложений необходимо выполнить определенные задачи по первоначальной настройке.

Эти задачи можно выполнить с помощью API (написав приложение PKCS #11) или интерфейса SMIT. Функции PKCS #11 SMIT можно вызвать, выбрав пункт **Управление подсистемой PKCS11** в главном меню или введя команду быстрого доступа **smit pkcs11**.

Инициализация маркера:

Каждый адаптер PKCS #11 (или маркер) необходимо инициализировать перед использованием.

В процессе инициализации маркеру присваивается уникальная метка. Эта метка однозначно идентифицирует маркер для приложений. По этой причине, метки не могут повторяться. Однако API не проверяет уникальность меток. Инициализация может быть выполнена приложением PKCS #11 или системным администратором с помощью SMIT. Если для маркера задан PIN системного администратора, то применяется значение по умолчанию - 87654321. Для обеспечения защиты подсистемы PKCS #11 это значение следует изменить после инициализации.

Для инициализации маркера выполните следующие действия:

1. Откройте меню управления маркерами, введя команду `smit pkcs11`.
2. Выберите пункт **Инициализировать маркер**.
3. В списке поддерживаемых адаптеров выберите адаптер PKCS #11.
4. Подтвердите выбранный адаптер, нажав клавишу Enter.

   Примечание: При этом будет удалена вся информация об этом маркере.
5. Введите PIN-код системного администратора (SO PIN) и уникальную метку маркера.

Если указан правильный PIN-код, то по завершении выполнения команды адаптер будет инициализирован.

Настройка PIN-кода системного администратора:

Ниже приведены инструкции по изменению значения PIN-кода SO по умолчанию.

Для изменения значения PIN-кода по умолчанию выполните следующие действия:

1. Введите `smit pkcs11`.
2. Выберите **Задать PIN-код системного администратора**.
3. Выберите инициализированный адаптер, для которого требуется задать PIN-код.
4. Введите текущий и новый PIN-коды.
5. Подтвердите новый PIN-код.

Инициализация пользовательского PIN-кода:

После инициализации маркера может потребоваться задать пользовательский PIN-код, чтобы приложения могли работать с объектами маркеров.

Обратитесь к документации устройства, чтобы определить, требуется ли идентификация пользователя на устройстве для работы с объектами.

Для инициализации пользовательского PIN-кода выполните следующие действия:

1. Откройте меню управления маркерами, введя команду `smit pkcs11`.
2. Выберите пункт **Инициализировать пользовательский PIN-код**.
3. В списке поддерживаемых адаптеров выберите адаптер PKCS #11.
4. Введите PIN-код системного администратора и пользовательский PIN-код.
5. Подтвердите пользовательский PIN-код.
6. После того, как вы подтвердите новое значение, пользовательский PIN-код будет изменен.

Сброс пользовательского PIN-кода:

Для сброса пользовательского PIN-кода можно либо повторно инициализировать его с помощью PIN-кода системного администратора, либо задать новое значение пользовательского PIN-код с помощью существующего.

Для сброса пользовательского PIN-кода выполните следующие действия:

1. Откройте меню управления маркерами, введя команду `smit pkcs11`.
2. Выберите пункт **Задать пользовательский PIN-код**.
3. Выберите инициализированный адаптер, для которого необходимо задать пользовательский PIN-код.
4. Введите текущий и новый пользовательские PIN-коды.
5. Подтвердите новый пользовательский PIN-код.

Применение стандарта шифрования с общим ключом #11

Для того чтобы приложение могло применять подсистему PKCS #11, должен быть запущен демон управления разъемами подсистемы, а приложение должно загрузить общий объект API.

Обычно демон управления разъемами запускается во время загрузки системы программой **inittab**, вызывающей сценарий `/etc/rc.pkcs11`. Перед запуском демона этот сценарий проверяет адаптеры, установленные в системе. Таким образом, демон управления разъемами будет недоступен, пока пользователь не войдет в систему. После запуска демона подсистема регистрирует все изменения в количестве и типе адаптеров без участия системного администратора.

API можно загрузить, подключив объект во время выполнения или воспользовавшись отложенным преобразованием символов. Например, приложение может получить список функций PKCS #11 следующим образом:

```
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);
```

Средства стандарта шифрования с общим ключом #11

Для управления криптографическими системами в операционной системе AIX применяются два средства: средство Управление ключом PKCS #11 и средство Администрирование PKCS #11. Получить доступ к этим средствам можно с помощью графического пользовательского интерфейса на основе Curses или интерфейса командной строки.

Примечание: Для доступа к средствам криптографической структуры AIX требуется применение функций пакетной обработки. Подробная информация об использовании функций пакетной обработки приведена в разделе “Пакетная обработка” на стр. 188.

Средство Управление ключом PKCS #11 является централизованным инструментом для работы с ключами, сертификатами и данными PKCS #11 в AIX. Объекты, управляемые этим инструментом, хранятся либо в поддерживаемых провайдерах PKCS #11, таких как семейство криптографических адаптеров IBM (например, IBM 4758, 4960 и 4764), или AIX Cryptographic Framework. С помощью средства Управление ключом PKCS #11 можно выполнять различные операции. Эти операции включают в себя создание Запроса на подпись

сертификата (CSR) PKCS #10 и генерацию собственных сертификатов. Кроме того, это средство можно использовать для поиска, просмотра, удаления, импорта, экспорта и резервного копирования данных объекта PKCS #11, а также для транспортировки данных объекта PKCS #11 между ключами PKCS #11. Графический пользовательский интерфейс средства можно запустить с помощью команды **p11km**. Средство загружает все доступные ключи PKCS #11. Можно просмотреть сведения об этих ключах с помощью клавиш стрелок или прокручивания списка ключей. Для выбора ключа выделите его с помощью клавиш стрелок и нажмите Enter. Командную строку средства можно запустить с помощью следующей команды:

```
p11km -b <пакетный-файл>
```

Средство Администрирование PKCS #11 является централизованным инструментом для управления Криптографической структурой PKCS #11 в AIX. Это средство позволяет администратору или системному администратору управлять ключами в AIX Cryptographic Framework. Это средство можно использовать для инициализации, создания и уничтожения ключей PKCS #11, управления замками, сброса паролей пользователей, подтверждения удаления объектов, указания надежных объектов, а также для выполнения тонкой настройки производительности и общего администрирования AIX Cryptographic Framework. Графический пользовательский интерфейс средства можно запустить с помощью команды **p11admin**. Средство загружает все доступные ключи PKCS #11. Можно просмотреть сведения об этих ключах с помощью клавиш стрелок или прокручивания списка ключей. Для выбора ключа выделите его с помощью клавиш стрелок и нажмите Enter. Командную строку средства можно запустить с помощью следующей команды:

```
p11admin -b
<пакетный-файл>
```

Профайлы команд:

Средства AIX Cryptographic Framework используют библиотеку OpenSSL для анализа файлов конфигурации, которые применяются при создании пользовательских профайлов. Эти профайлы можно использовать для установки атрибутов, таких как цвета графического пользовательского интерфейса, в команде **p11km** и команде **p11admin**.

С помощью формата файла, который описан в разделе “Пакетная обработка” на стр. 188, можно создать и изменить следующие файлы profile для настройки графического пользовательского интерфейса.

Примечание: После создания файлов profile, присвойте им имя и сохраните в домашнем каталоге следующим образом:

```
$HOME/.p11km

$HOME/.p11admin
```

Поддерживаются следующие атрибуты цвета графического пользовательского интерфейса:

```
action_name = "GUI_COLORS"
gui_fg_color = "<имя цвета>" ## Цвет текста
gui_bg_color = "<имя цвета>" ## Цвет фона
gui_vc_color = "<имя цвета>" ## Цвет содержимого
```

Где <*имя цвета*> - это одно из следующих значений:

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

Пример: профайл p11km ($HOME/.p11km)

```
[p11km_cmd]
gui_fg_color = "RED"
gui_bg_color = "BLACK"
gui_vc_color = "WHITE"
```

Пример: профайл p11admin ($HOME/.p11admin)

```
[p11admin_cmd]
gui_fg_color = "BLUE"
gui_bg_color = "LIGHT GRAY"
gui_vc_color = "BLACK"
```

Пакетная обработка:

Из командной строки можно выполнить команды пакетной обработки для тех же задач, которые доступны в графическом пользовательском интерфейсе средств PKCS #11.

Формат команды для средства Управление ключом PKCS #11 (p11km):

```
p11km -b <пакетный-файл>
```

Формат команды для средства Администрирование ключа PKCS #11 (p11admin):

```
p11admin -b <пакетный-файл>
```

Так как эти средства используют библиотеку OpenSSL для синтаксического анализа пакетных файлов, формат пакетных файлов соответствует типичному формату файла конфигурации OpenSSL. Каждый раздел является отдельной командой, и пара атрибута и значения предоставляют информацию, требуемую для обработки. Каждая команда раздела обрабатывается в пакете сверху вниз. Если отдельная пакетная команда не выполнена, выдается ошибка и пакетная обработка завершается без выполнения последующих команд.

Ниже приведен пример формата файла конфигурации OpenSSL.

```
[раздел1]
атрибут1 = "значение1"
атрибут2 = "значение2"
...
атрибутN = "значениеN"
[раздел2]
атрибут1 = "значение1"
атрибут2 = "значение2"
...
атрибутN = "значениеN"
...
...
[разделN]
атрибут1 = "значение1"
атрибут2 = "значение2"
...
атрибутN = "значениеN"
```

Для того чтобы обеспечить сосуществование разделов команд средства PKCS #11 с разделами файла конфигурации OpenSSL, используйте следующие префиксы для разделов PKCS #11:

средство p11km
p11km_cmd

средство p11admin
p11admin_cmd

Каждый раздел p11km_cmd или p11admin_cmd должен содержать только один атрибут action_name со строковым значением, обозначающий определенную команду, связанную с разделом. Простейший пример - это файл, который содержит один раздел команды, не имеющей дополнительных параметров. Ниже приведен пример использования средства p11km для выполнения пакетной команды, которая перечисляет доступные ключи PKCS #11 в системе:

```
[p11km_cmd_list_my_tokens]
action_name="LIST_TOKENS"
```

Каждая команда пакета поддерживает необязательный булевский атрибут:

```
start_gui="<boolean>"
```

При выполнении пакетной команды, которая содержит булевский атрибут со значением TRUE, обработка пакета завершается после выполнения этой команды, и запускается графический пользовательский интерфейс.

Примечание: Если пакетный файл содержит команду, которая включает в себя необязательный атрибут **start_gui**, все перечисленные далее пакетные команды не обрабатываются.

Пакетные команды:

Пакетные команды предоставляют доступ через командную строку к средствам PKCS #11.

Следующие пакетные команды доступны в средстве Управление ключом PKCS #11 (p11km).

Примечание: Для использования пакетных команд выполните следующие действия:

1. Создайте и измените пакетный файл, как описано в разделе “Пакетная обработка” на стр. 188.
2. Создайте новые разделы p11km_cmd, содержащие атрибуты для пакетных команд, которые вы хотите использовать.

Перечислить доступные ключи PKCS #11
Генерирует отчет и показывает ключ и информацию о замке для доступных ключей PKCS #11.

Требуемые атрибуты
```
action_name = "LIST_TOKENS"
```

Необязательные атрибуты
```
start_gui = "<boolean>"
```

Где <*boolean*> - это или TRUE, или FALSE

Пример
```
[p11km_cmd_list_tokens]
action_name = "LIST_TOKENS"
```

Перечислить доступные механизмы PKCS #11
Генерирует отчет и показывает доступные механизмы PKCS #11, поддерживаемые определенным ключом PKCS #11 (задается с помощью значений атрибута замка и драйвера).

Требуемые атрибуты

```
action_name = "LIST_MECHANISMS"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
```

Где <*номер замка*> - это положительное целое число, а <*имя драйвера*> - это одно из следующих значений:

Значение	Описание
AIX	AIX OS Cryptographic Framework
IBM_4758_4960	IBM 4758/4960 Cryptographic Hardware Adapters
IBM_4764	IBM 4764 Cryptographic Hardware Adapter
Other	Если указано значение OTHER, необходимо также задать атрибут **p11_driver_path**.

Необязательные атрибуты

```
start_gui = "<boolean>"
```

Дополнительные атрибуты

```
p11_driver_path = "<путь к драйверу PKCS#11>"
```

Где <*путь к драйверу PKCS#11*> - это полный путь и имя файла UNIX библиотеки PKCS #11, которая используется для команды. Этот атрибут может быть задан, только когда значение атрибута **p11_driver** равно OTHER.

Пример

```
[p11km_cmd_list_4764_slot_0_mechs]
action_name = "LIST_MECHANISMS"
p11_driver = "IBM_4764"
p11_slot = "0"
start_gui = "TRUE"
```

Перечислить доступные объекты PKCS #11

Генерирует отчет и показывает доступные объекты PKCS #11, поддерживаемые ключом PKCS #11 (задается с помощью значений атрибута замка и драйвера).

Требуемые атрибуты

```
action_name = "LIST_OBJECTS"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
```

Необязательные атрибуты

```
p11_login = "<boolean>"
p11_label = "<string>"
p11_class = "<Класс объекта PKCS#11>"
p11_private = "<boolean>"
p11_trusted = "<boolean>"
p11_sensitive = "<boolean>"
start_gui = "<boolean>"
```

Где <*Класс объекта PKCS#11*> - это одно из следующих значений, как определено в спецификации PKCS #11 из 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
```

Пример

```
[p11km_cmd_list_private_objs]
action_name = "LIST_OBJECTS"
p11_login = "TRUE"
p11_private = "TRUE"
p11_driver = "AIX"
p11_slot = "5"
```

Изменить PIN пользователя ключа PKCS #11:

Изменяет PIN пользователя ключа PKCS #11, который используется при входе в ключ.

Требуемые атрибуты

```
action_name = "CHANGE_USER_PIN"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
```

Необязательные атрибуты

```
start_gui = "<boolean>"
```

Пример

```
[p11km_cmd_change_my_pin]
action_name = "CHANGE_USER_PIN"
p11_slot = "1337"
p11_driver = "IBM_4764"
```

Удалить объекты PKCS #11

Удаляет объекты PKCS #11. Объекты удаляются на основании нумерованного списка объектов, который выдается в результате выполнения команды **LIST_OBJECTS** и использования того же шаблона со следующими атрибутами:

```
p11_label = "<string>"
p11_class = "<Класс объекта PKCS#11>"
p11_private = "<boolean>"
p11_trusted = "<boolean>"
p11_sensitive = "<boolean>"
p11_login = "<boolean>"
```

Внимание: Так как состояние и совместимость ключа не поддерживается между пакетными процессами, объекты могут быть непреднамеренно удалены. Порядок перечисленных объектов изменяется при добавлении и удалении объектов другими процессами, которые выполняются для того же ключа между моментом первоначального указания объекта в перечислении и моментом его удаления.

Требуемые атрибуты

```
action_name = "DELETE_OBJECTS"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
p11_objects = "<CSV>"
```

Где <*CSV*> - это или слово ALL (все объекты ключа), или список через запятую положительных целых чисел, которые соответствуют объектам в нумерованном порядке появления, использующем следующие необязательные атрибуты.

Необязательные атрибуты

```
p11_label = "<string>"
p11_class = "<Класс объекта PKCS#11>"
p11_private = "<boolean>"
p11_trusted = "<boolean>"
p11_sensitive = "<boolean>"
p11_login = "<boolean>"
start_gui = "<boolean>"
```

Пример

```
[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"
```

Переместить объекты PKCS #11:

Перемещает объекты PKCS #11. Объекты перемещаются на основании нумерованного списка объектов, который выдается в результате выполнения команды **LIST_OBJECTS** и использования того же шаблона.

Внимание: Так как состояние и совместимость ключа не поддерживается между пакетными процессами, объекты могут быть непреднамеренно перемещены. Порядок перечисленных объектов изменяется при добавлении и удалении объектов другими процессами, которые выполняются для того же ключа между моментом первоначального указания объекта в перечислении и моментом его перемещения.

Требуемые атрибуты

```
action_name = "MOVE_OBJECTS"
########################################
###### Идентификация исходного ключа: ######
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
########################################
###### Идентификация целевого ключа: ######
p11_driver_target = "<имя драйвера>"
p11_slot_target = "<номер замка>"
########################################
###### Объекты, перемещаемые в назначение: ######
p11_objects = "<CSV>"
```

Необязательные атрибуты

```
p11_label = "<string>"
p11_class = "<Класс объекта PKCS#11>"
p11_private = "<boolean>"
p11_trusted = "<boolean>"
p11_sensitive = "<boolean>"
p11_login = "<boolean>"
start_gui = "<boolean>"
```

Пример

```
[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"
```

Скопировать объекты PKCS #11

Копирует объекты PKCS #11. Объекты копируются на основании нумерованного списка объектов, который выдается в результате выполнения команды **LIST_OBJECTS** и использования того же шаблона.

Внимание: Так как состояние и совместимость ключа не поддерживается между пакетными процессами, объекты могут быть непреднамеренно скопированы. Порядок перечисленных объектов изменяется при добавлении и удалении объектов другими процессами, которые выполняются для того же ключа между моментом первоначального указания объекта в перечислении и моментом его копирования.

Требуемые атрибуты

```
action_name = "COPY_OBJECTS"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
p11_driver_target = "<имя драйвера>"
p11_slot_target = "<номер замка>"
p11_objects = "<CSV>"
```

Необязательные атрибуты

```
p11_label = "<string>"
p11_class = "<Класс объекта PKCS#11>"
p11_private = "<boolean>"
p11_trusted = "<boolean>"
p11_sensitive = "<boolean>"
p11_login = "<boolean>"
start_gui = "<boolean>"
```

Пример

```
[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" ## ТРЕБУЕТСЯ ДЛЯ MGT ЧАСТНОГО ОБЪЕКТА.
```

Экспортировать и создать резервную копию объектов PKCS #11 в файле

Экспортирует и создает резервную копию объектов PKCS #11. Объекты экспортируются и для них создается резервная копия на основании нумерованного списка объектов, который выдается в результате выполнения команды **LIST_OBJECTS** и использования того же шаблона.

Внимание: Так как состояние и совместимость ключа не поддерживается между пакетными процессами, объекты могут быть непреднамеренно экспортированы. Порядок перечисленных объектов изменяется при добавлении и удалении объектов другими процессами, которые выполняются для того же ключа между моментом первоначального указания объекта в перечислении и моментом его экспорта.

Требуемые атрибуты

```
action_name = "EXPORT_OBJECTS"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
p11_object_file = "<имя файла>"
p11_objects = "<CSV>"
```

Необязательные атрибуты

```
p11_label = "<string>"
p11_class = "<Класс объекта PKCS#11>"
p11_private = "<boolean>"
p11_trusted = "<boolean>"
p11_sensitive = "<boolean>"
p11_login = "<boolean>"
start_gui = "<boolean>"
```

Пример

```
[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"
```

Импортировать объекты PKCS #11 из файла

Импортирует объекты PKCS #11, которые были созданы из файла экспорта PKCS #11.

Требуемые атрибуты

```
action_name = "IMPORT_OBJECTS"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
p11_object_file = "<имя файла>"
```

Необязательные атрибуты

```
p11_login = "<boolean>" # ТРЕБУЕТСЯ ДЛЯ ИМПОРТА ЛЮБЫХ ЧАСТНЫХ ОБЪЕКТОВ
start_gui = "<boolean>"
```

Пример

```
[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"
```

Создать собственный сертификат

Создает собственный сертификат X.509 и связанные объекты PKCS #11 в ключе PKCS #11.

Требуемые атрибуты

```
action_name = "CREATE_SSC"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
p11_login = "TRUE"
p11_ssc_label = "<string>"
p11_ssc_config = "<файл конфигурации openssl>"
```

Где Where <*файл конфигурации openssl*> - это полный путь и имя файла UNIX для файла конфигурации OpenSSL, который заполнен значениями, использованными при создании собственного сертификата.

Необязательные атрибуты

```
start_gui = "<boolean>"
```

Пример

```
[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"
```

Создать запрос на подписание сертификата PKCS #10

Создает запрос сертификации или запрос на подписание сертификата (CSR) PKCS #10.

Требуемые атрибуты

```
action_name = "CREATE_CSR"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
p11_login = "TRUE"
p11_csr_label = "<string>"
p11_csr_file = "<путь файла вывода CSR>"
p11_csr_type = "<DER или Base64>'
p11_csr_config = "<файл конфигурации openssl>"
```

Где <*DER или Base64*> или генерирует файл вывода CSR, зашифрованного в ASN.1 (DER), или файл вывода CSR, зашифрованного в Base64, а <*путь к файлу вывода CSR*> ссылается на полный путь и имя файла UNIX вывода CSR.

Необязательные атрибуты

```
start_gui = "<boolean>"
```

Пример

```
[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"
```

Следующие пакетные команды доступны в средстве Администрирование PKCS #11 (p11admin).

Примечание: Для использования пакетных команд выполните следующие действия:

1. Создайте и измените пакетный файл, как описано в разделе “Пакетная обработка” на стр. 188.
2. Создайте новые разделы p11km_cmd, содержащие атрибуты для пакетных команд, которые вы хотите использовать.

Перечислить доступные ключи PKCS #11

Генерирует отчет и показывает ключ и информацию о замке для доступных ключей PKCS #11.

Требуемые атрибуты

```
action_name = "ADM_LIST_TOKENS"
```

Необязательные атрибуты

```
start_gui = "<boolean>"
```

Где <*boolean*> - это или TRUE, или FALSE

Пример

```
[p11admin_cmd_list_tokens]
action_name = "ADM_LIST_TOKENS"
```

Перечислить доступные механизмы PKCS #11

Генерирует отчет и показывает доступные механизмы PKCS #11, поддерживаемые ключом PKCS #11 (задается с помощью значений атрибута замка и драйвера).

Требуемые атрибуты

```
action_name = "ADM_LIST_MECHANISMS"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
```

Где <*номер замка*> - это положительное целое число, а <*имя драйвера*> - это одно из следующих значений:

Значение	Описание
AIX	AIX OS Cryptographic Framework
IBM_4758_4960	IBM 4758/4960 Cryptographic Hardware Adapters
IBM_4764	IBM 4764 Cryptographic Hardware Adapter
Другие	Если указано значение OTHER, необходимо также задать атрибут **p11_driver_path**.

Необязательные атрибуты

```
start_gui = "<boolean>"
```

Дополнительные атрибуты

```
p11_driver_path = "<путь к драйверу
PKCS#11>"
```

Где <*путь к драйверу PKCS#11*> - это полный путь и имя файла UNIX библиотеки PKCS #11, которая используется для команды. Этот атрибут может быть задан, только когда значение атрибута **p11_driver** равно OTHER.

Пример

```
[p11admin_cmd_list_4764_slot_0_mechs]
action_name = "ADM_LIST_MECHANISMS"
p11_driver = "IBM_4764"
p11_slot = "0"
start_gui = "TRUE"
```

Показать информацию для ключа PKCS #11

Показывает информацию о ключе и замке PKCS #11 для ключа PKCS #11.

Требуемые атрибуты

```
action_name = "ADM_SHOW_TOKEN_INFO"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
```

Необязательные атрибуты

```
start_gui = "<boolean>"
```

Пример

```
[p11admin_cmd]
action_name = "ADM_SHOW_TOKEN_INFO"
p11_slot = "411"
p11_driver = "IBM_4764"
```

Инициализировать ключ PKCS #11:

Инициализирует ключ PKCS #11. Инициализация сбрасывает ключ, стирает все сохраненные объекты и данные PKCS#11 и позволяет повторно пометить ключ.

Внимание: Так как все объекты и данные PKCS #11 стираются в процессе инициализации, убедитесь в том, что они не нужны, перед тем как инициализировать ключ PKCS #11.

Требуемые атрибуты

```
action_name = "ADM_INIT_TOKEN"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>" ## СОВПАДАЕТ С 'p11_init_slot'
p11_init_slot = "<номер замка>" ## СОВПАДАЕТ С 'p11_slot'
p11_init_label = "<string>" ## НОВАЯ МЕТКА КЛЮЧА
```

Необязательные атрибуты

```
start_gui = "<boolean>"
```

Пример

```
[p11admin_cmd]
action_name = "ADM_INIT_TOKEN"
p11_slot = "1"
p11_driver = "IBM_4764"
p11_init_slot = "1"
p11_init_label = "ABC Token"
```

Просмотреть таймер для ключа PKCS #11

Показывает аппаратный таймер для ключа PKCS #11, если этот ключ имеет таймер.

Требуемые атрибуты

```
action_name = "ADM_CLOCK_VIEW"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
```

Необязательные атрибуты

```
start_gui = "<boolean>"
```

Пример

```
[p11admin_cmd]
action_name = "ADM_CLOCK_VIEW"
p11_slot = "1"
p11_driver = "IBM_4764"
```

Установить таймер для ключа PKCS #11

Устанавливает аппаратный таймер для ключа PKCS #11, если этот ключ имеет таймер.

Требуемые атрибуты

```
action_name = "ADM_CLOCK_SET"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
p11_clock_set = "<данные таймера>"
```

Где <*данные таймера*> - это текущие дата и время UTC в следующем формате: ЧЧ:ММ:СС мм-дд-ГГГГ.

Необязательные атрибуты

```
start_gui = "<boolean>"
```

Пример

```
[p11admin_cmd]
action_name = "ADM_CLOCK_SET"
p11_slot = "1"
p11_driver = "IBM_4764"
p11_clock_set = "23:59:59 12-31-1999"
```

Сбросить PIN для пользователя ключа PKCS #11

Сбрасывает PIN для пользователя ключа PKCS #11

Требуемые атрибуты

```
action_name = "ADM_RESET_USER_PIN"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
```

Необязательные атрибуты

```
start_gui = "<boolean>"
```

Пример

```
[p11admin_cmd_change_so_pin]
action_name = "ADM_RESET_USER_PIN"
p11_driver = "AIX"
p11_slot = "0"
```

Изменить PIN для системного администратора ключа PKCS #11

Изменяет PIN для системного администратора ключа PKCS #11. Этот PIN используется при выполнении администрирования ключа.

Требуемые атрибуты

```
action_name = "ADM_CHANGE_SO_PIN"
p11_driver = "<имя драйвера>"
p11_slot = "<номер замка>"
```

Необязательные атрибуты

```
start_gui = "<boolean>"
```

Пример

```
[p11admin_cmd_change_so_pin]
action_name = "ADM_CHANGE_SO_PIN"
p11_slot = "888"
p11_driver = "IBM_4764"
```

Встраиваемые модули идентификации

Архитектура встраиваемых модулей идентификации (PAM) предоставляет системным администраторам возможность встраивать в существующую систему несколько механизмов защиты с помощью встраиваемых модулей.

Приложения с поддержкой PAM могут *встраиваться* в существующие системы, не требуя изменения уже работающих приложений. Такая гибкость предоставляет администраторам следующие возможности:

- Выбирать для приложения системную службу идентификации
- Применять с данной службой несколько механизмов идентификации
- Добавлять новые модули службы идентификации без изменения существующих приложений
- Применять указанные ранее пароли для идентификации с помощью различных модулей

PAM состоит из библиотеки, встраиваемых модулей и файла конфигурации. Библиотека PAM реализует интерфейс прикладных программ (API) PAM и служит для управления транзакциями PAM и вызова служебных интерфейсов (SPI) PAM, определенных во встраиваемых модулях. Встраиваемые модули динамически загружаются библиотекой на основании вызывающих служб и соответствующих им записей файла конфигурации. Успешность идентификации определяется не только встраиваемым модулем, но и алгоритмом работы службы. Применение концепции *стека* позволяет настроить службу для применения нескольких методов идентификации. Модули также можно настроить на применение уже введенного пароля, без повторного запроса пароля у пользователя.

Администратор может настроить систему AIX для работы с PAM, изменив атрибут **auth_type** в разделе usw файла `/etc/security/login.cfg`. Значение `auth_type = PAM_AUTH` позволяет настроить команды PAM таким образом, чтобы для идентификации вызывались не традиционные функции AIX, а API PAM. Этот параметр можно изменять динамически. Изменение значения вступает в силу без перезагрузки системы. Дополнительная информация об атрибуте **auth_type** приведена в описании файла `/etc/security/login.cfg`. Для распознавания атрибута **auth_type** и получения возможности идентификации с помощью PAM были изменены следующие стандартные команды и приложения AIX:

- **login**
- **passwd**
- **su**
- **ftp**
- **telnet**
- **rlogin**
- **rexec**
- **rsh**
- **snappd**
- **imapd**
- **dtaction**
- **dtlogin**
- **dtsession**

Ниже проиллюстрировано взаимодействие между приложениями, библиотекой PAM, файлом конфигурации и модулями PAM в системе, настроенной для использования PAM. Приложения с поддержкой PAM, вызывают API PAM из библиотеки PAM. На основании указанной в файле конфигурации записи приложения и вызовов SPI PAM в модуле библиотека определяет модуль для загрузки. Модуль PAM взаимодействует с приложением с помощью функции ведения диалога, реализованной в приложении. Затем, в зависимости от успеха или неудачи выполненной модулем проверки, а также в зависимости от алгоритма, определенного в файле конфигурации, определяется необходимость загрузки следующего модуля. Если такая необходимость есть, то обработка продолжается; в противном случае результат возвращается приложению.

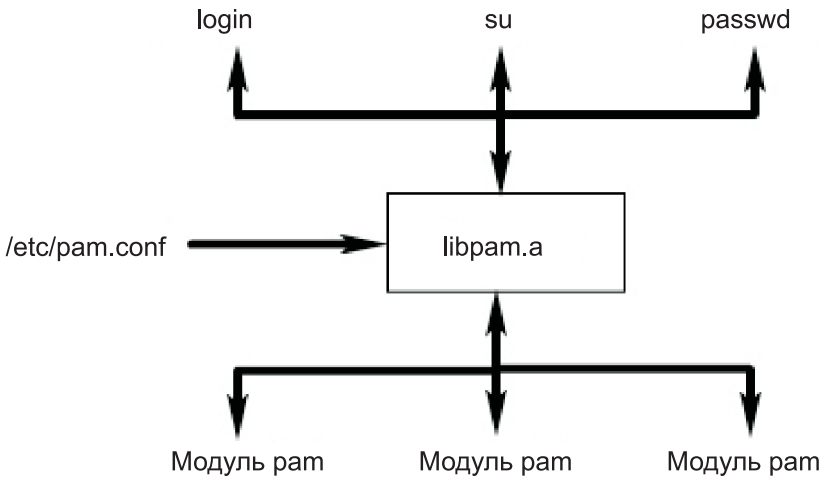


Рисунок 3. Среда и объекты PAM. На рисунке показаны команды, с помощью которых команды с поддержкой PAM обращаются через библиотеку PAM к соответствующим модулям.

Библиотека PAM

Библиотека PAM `/usr/lib/libpam.a` содержит API PAM, выполняющий функцию общего интерфейса для всех приложений PAM и управляющий загрузкой модулей.

Модули загружаются библиотекой PAM на основании алгоритма, определенного в файле `/etc/pam.conf`

Следующие функции API PAM вызывают соответствующие SPI PAM, реализованные в модулях PAM. Например, API pam_authenticate вызывает в модуле PAM SPI pam_sm_authenticate.

- `pam_authenticate`
- `pam_setcred`
- `pam_acct_mgmt`
- `pam_open_session`
- `pam_close_session`
- `pam_chauthtok`

Кроме того, в библиотеке PAM предусмотрен ряд структурных API, позволяющих приложению вызывать модули PAM и передавать им информацию. В следующей таблице перечислены структурные API библиотеки PAM, включенные в AIX и их функции:

API среды PAM	**Функция**
pam_start	Установить сеанс PAM
pam_end	Прервать сеанс PAM
pam_get_data	Получить данные для конкретного модуля
pam_set_data	Задать данные для конкретного модуля
pam_getenv	Получить значение определенной переменной среды PAM
pam_getenvlist	Получить список всех определенных переменных среды PAM и их значений
pam_putenv	Задать переменную среды PAM
pam_get_item	Получить общую информацию PAM
pam_set_item	Задать общую информацию PAM
pam_get_user	Получить имя пользователя
pam_strerror	Получить стандартное сообщение об ошибке PAM

Модули PAM

Модули PAM позволяют совместно или независимо применять в системе несколько механизмов идентификации.

Каждый модуль PAM должен реализовать функции по крайней мере одного из четырех типов. Типы модулей описаны ниже с указанием соответствующих SPI PAM, которые обязательно должны присутствовать в модуле каждого типа.

Модули идентификации
Предназначены для идентификации пользователей, а также создания, обновления и уничтожения одноразовых разрешений. Такие модули идентифицируют пользователей с помощью предоставленных идентификационных данных.

Функции модулей идентификации:

- pam_sm_authenticate
- pam_sm_setcred

Модули управления учетными записями
Проверяют допустимость учетных записей пользователей после идентификации пользователя модулем идентификации. Обычно выполняемая этими модулями проверка включает проверку срока действия учетной записи и ограничений на пароль.

Функции модуля управления учетными записями:

- pam_sm_acct_mgmt

Модули управления сеансом
Инициируют и прерывают сеансы пользователей. Кроме того, должна быть обеспечена поддержка контроля сеанса.

Функции модулей управления сеансом:

- pam_sm_open_session
- pam_sm_close_session

Модули управления паролями
Обеспечивают изменение пароля и проверку связанных с ним атрибутов.

Функции модулей управления паролями:

- pam_sm_chauthtok

Файл конфигурации PAM

Файл конфигурации `/etc/pam.conf` содержит записи для каждого типа модулей PAM и служит для определения алгоритма обработки служебных запросов с помощью заданной последовательности модулей.

Записи файла конфигурации включают следующие поля, разделяемые пробелами или символами табуляции:

имя-службы тип-модуля управляющий-флаг полное-имя-модуля опции-модуля

Ниже приведены описания этих полей:

имя-службы
Задает имя службы. Для определения модуля по умолчанию, используемого приложениями, которые не указаны в записи, применяется ключевое слово OTHER.

тип-модуля
Задает тип модуля. Допустимы следующие типы: **auth**, **account**, **session** и **password**. Модуль обеспечивает поддержку одного или нескольких типов.

управляющий-флаг
Задает способ вызова модуля. Поддерживаются следующие флаги: required, requisite, sufficient и optional.

полное-имя-модуля

Задает модуль, загружаемый для службы. Допустимыми значениями для *путь-к-модулю* являются полный путь к модулю или только имя модуля. Если задан полный путь к модулю, библиотека PAM использует этот *путь-к-модулю* для загрузки 32-разрядных служб или использует 64 подкаталога для 64-разрядных служб. Если полный путь к модулю не задан, то библиотека PAM добавляет к имени модуля префикс `/usr/lib/security` (для 32-разрядных служб) или `/usr/lib/security/64` (для 64-разрядных служб).

опции-модуля

Задает ограниченный по объему список опций для передачи модулям. Допустимые в этом поле значения зависят от того, какие опции поддерживаются модулем, заданным в поле *полное-имя-модуля*. Это необязательное поле.

Записи, указанные в неправильном формате или содержащие неправильные значения в полях **тип-модуля** или **управляющий-флаг**, игнорируются библиотекой PAM. Записи, начинающиеся с символа решетки (#), считаются комментариями и игнорируются.

PAM поддерживает принцип стека, позволяющий службе применять несколько механизмов. Стек реализуется путем создания в файле конфигурации нескольких записей с одинаковым значением поля **тип-модуля**. Модули вызываются в том порядке, в котором они перечислены в файле для отдельной службы; результат вызова определяется по значению поля **управляющий-флаг** для каждой записи. Ниже перечислены допустимые значения поля **управляющий-флаг** с указанием соответствующих действий, выполняемых в стеке:

Значение поля control_flag	**Поведение**
required	Обязательный. Для получения положительного результата проверка должна быть успешно выполнена всеми обязательными модулями. Если от одного или нескольких обязательных модулей будет получено сообщение об отрицательном результате, то будет предпринята попытка проверки с помощью всех обязательных модулей, но будет возвращено первое полученное сообщение об ошибке.
requisite	Необходимый. Аналогично значению required, за исключением того, что если от необходимого модуля будет получено сообщение об отрицательном результате, следующие модули в стеке не выполняются, и немедленно возвращается первый код ошибки от обязательного или необходимого модуля.
sufficient	Достаточный. Если модуль, помеченный как достаточный, успешно провел проверку, и при этом все предыдущие обязательные и достаточные модули не обнаружили ошибок, то все остальные модули игнорируются и возвращается сообщение о положительном результате проверки.
optional	Необязательный. Если в стеке нет обязательных модулей и ни один из достаточных модулей не вернул сообщение об успешной проверке, то для получения положительного результата необходимо, чтобы хотя бы один из необязательных модулей успешно провел проверку. Если какой-либо другой модуль вернул сообщение об успешной проверке, то отрицательный результат необязательного модуля игнорируется.

Следующий фрагмент файла `/etc/pam.conf` является примером стека модулей типа `auth` для службы входа в систему.

```
#
# Файл конфигурации PAM /etc/pam.conf
#

# Управление идентификацией
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
```

Пример файла конфигурации содержит три записи для службы входа в систему. Поскольку модули `pam_ckfile` и `pam_aix` определены как обязательные, для успешного выполнения операции оба модуля должны отработать успешно. Третий модуль `pam_test` указан как необязательный. Результат проверки с

помощью этого модуля не повлияет на возможность входа пользователя в систему. Опция use_first_pass модуля pam_test требует использовать пароль, введенный при обращении к предыдущему модулю, не запрашивая у пользователя новый пароль.

Указание в качестве имени службы ключевого слова OTHER позволяет задать значения по умолчанию для всех служб, не указанных явно в файле конфигурации. Настройка значений по умолчанию гарантирует, что в любой ситуации будет вызван по крайней мере один модуль данного типа. В данном случае все службы, кроме службы входа в систему, не будут работать, поскольку все вызовы модуля pam_prohibit возвращают ошибку PAM.

Модуль pam_aix

Модуль pam_aix - это модуль PAM, обеспечивающий приложениям с поддержкой PAM доступ к службам защиты AIX с помощью интерфейсов, вызывающих службы, эквивалентные службам AIX (если таковые есть).

Такие службы реализуются с помощью загружаемых модулей идентификации или встроенной функции AIX на основе пользовательских определений и файлов конфигурации methods.cfg. Коды ошибок, сформированные службой AIX, преобразуются в соответствующие коды ошибок PAM.

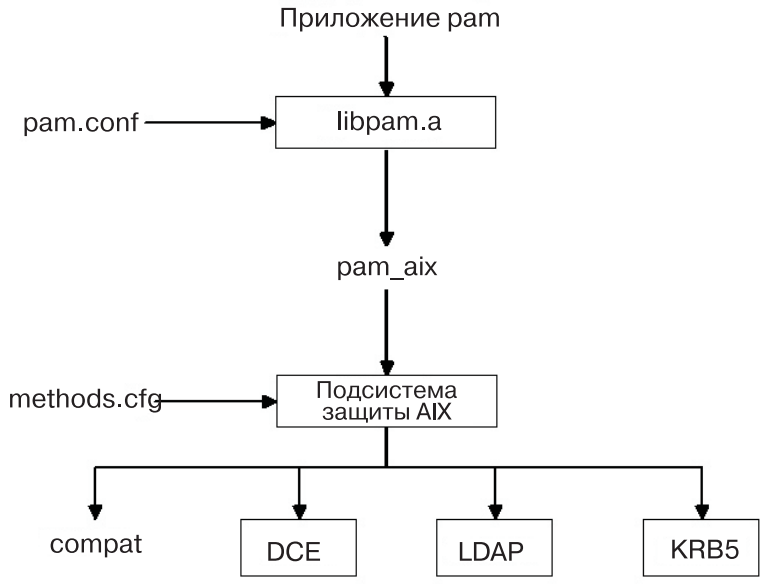


Рисунок 4. Обращение приложений PAM к подсистеме защиты AIX

На этом рисунке показан способ обработки вызова API в приложении PAM в том случае, если в файле /etc/pam.conf настроено применение модуля pam_aix. Как показано на рисунке, интеграция позволяет выполнить идентификацию пользователей с помощью любых загружаемых модулей (DCE, LDAP или KRB5), либо с помощью файлов AIX (совм).

Модуль pam_aix устанавливается в каталог /usr/lib/security. Интеграция модуля pam_aix требует применения этого модуля в файле /etc/pam.conf. В следующем примере файла /etc/pam.conf стек не применяется, хотя это и допустимо:

```
#
# Управление идентификацией
#
OTHER   auth     required       /usr/lib/security/pam_aix

#
# Управление учетными записями
#
```

```
OTHER   account  required       /usr/lib/security/pam_aix

#
# Управление сеансами
#
OTHER   session  required       /usr/lib/security/pam_aix

#
# Управление паролями
#
OTHER   password required       /usr/lib/security/pam_aix
```

В модуле pam_aix реализованы функции SPI pam_sm_authenticate, pam_sm_chauthok и pam_sm_acct_mgmt. Функции pam_sm_setcred, pam_sm_open_session и pam_sm_close_session также реализованы в модуле pam_aix, но эти они возвращают результат PAM_SUCCESS.

Ниже приведена приблизительная таблица преобразования вызовов SPI PAM в вызовы подсистемы защиты AIX:

```
 PAM SPI                    AIX
=========                   =====
pam_sm_authenticate   -->  authenticate
pam_sm_chauthtok      -->  passwdexpired, chpass
                           Примечание: passwdexpired проверяется только
                           при передаче флага
                           PAM_CHANGE_EXPIRED_AUTHTOK.
pam_sm_acct_mgmt      -->  loginrestrictions, passwdexpired
pam_sm_setcred        -->  Нет соответствующего вызова, возвращается PAM_SUCCESS
pam_sm_open_session   -->  Нет соответствующего вызова, возвращается PAM_SUCCESS
pam_sm_close_session  -->  Нет соответствующего вызова, возвращается PAM_SUCCESS
```

Данные, предназначенные для передачи подсистеме защитыAIX, могут быть заданы либо с помощью функции pam_set_item перед обращением к модулю, либо, если данных еще нет, то их может запросить pam_aix.

Загружаемый модуль идентификации PAM

Службы защиты AIX можно настроить таким образом, чтобы модули PAM вызывались с помощью существующей среды загружаемых модулей идентификации AIX.

При правильной настройке файла /usr/lib/security/methods.cfg загружаемый модуль PAM передает запросы служб защиты AIX (passwd, login, и т.п.) в библиотеку PAM. Библиотека PAM обращается к файлу /etc/pam.conf, определяет, какой модуль PAM следует применить, а затем вызывает соответствующий SPI PAM. Значения возврата PAM преобразуются в коды ошибок AIX и возвращаются в вызывающую программу.

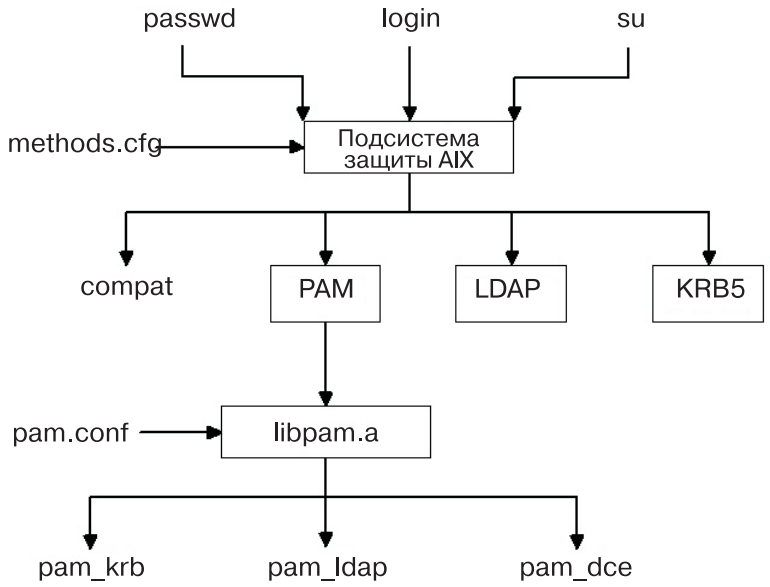


Рисунок 5. Взаимодействие служб защиты AIX и модуля PAM

На этом рисунке показана последовательность обработки вызова служб защиты AIX при правильной настройке PAM. Показанные модули PAM (pam_krb, pam_ldap и pam_dce) представляют собой примеры решений, созданных независимыми разработчиками.

Загрузочный модуль PAM устанавливается в каталог `/usr/lib/security` и применяется только для идентификации. Для создания составного загружаемого модуля модуль PAM должен применяться вместе с базой данных. В следующем примере показаны разделы, которые можно добавить в файл `methods.cfg` для создания составного модуля PAM, использующего базу данных files. Ключевое слово BUILTIN, указанное в атрибуте **db**, означает, что база данных представляет собой файлы UNIX.

```
PAM:
        program = /usr/lib/security/PAM

PAMfiles:
        options = auth=PAM,db=BUILTIN
```

Создание и изменение пользователей осуществляется с помощью опции -R команд администрирования, а также с помощью указания атрибута *SYSTEM* при создании пользователя. Например:

```
mkuser -R PAMfiles SYSTEM=PAMfiles registry=PAMfiles pamuser
```

Таким образом, все дальнейшие вызовы служб защиты AIX (login, passwd, и т.п.) получают информацию о выполнении идентификации с помощью модуля PAM. В данном примере применялась база данных **files**, но могут использоваться и другие установленные базы данных, например, LDAP. Создание пользователей описанным выше способом приведет к следующему преобразованию служб защиты AIX в вызовы API PAM:

```
 AIX                     PAM API
=====                   =========
authenticate       -->  pam_authenticate
chpass             -->  pam_chauthtok
passwdexpired      -->  pam_acct_mgmt
passwdrestrictions -->  Нет соответствующего вызова, возвращается сообщение об успешной проверке
```

Настройка файла `/etc/pam.conf` позволяет направлять вызовы API PAM различным модулям PAM для идентификации. Для более гибкой настройки механизма идентификации можно реализовать стек.

Данные, запрошенные службами защиты AIX, передаются PAM через функцию pam_set_item, поскольку PAM не взаимодействует непосредственно с пользователями. Модули PAM, обеспечивающие интеграцию с модулем PAM, должны получать данные посредством вызовов pam_get_item и не должны пытаться запрашивать ввод непосредственно у пользователя.

Средства обнаружения циклов позволяют найти такие ошибки конфигурации, как: службы защиты AIX передают запрос PAM, а модуль PAM, в свою очередь, пытается вызвать службы защиты AIX. Обнаружение такого цикла приведет к немедленной выдаче сообщения об ошибке.

Примечание: Файл `/etc/pam.conf` *не* должен использовать модуль pam_aix при вызове модуля PAM из службы защиты AIX, поскольку это приведет к зацикливанию.

Добавление модуля PAM

Для включения механизмов множественной идентификации можно добавить модуль PAM.

1. Скопируйте 32-разрядную версию модуля в каталог `/usr/lib/security`; 64-разрядная версия копируется в каталог `/usr/lib/security/64`.
2. Укажите в качестве владельца файла пользователя `root`, а в качестве режима доступа - значение `555`. Библиотека PAM не загружает модули, если эти модули принадлежат не пользователю root.
3. Обновите файл конфигурации `/etc/pam.conf`, включив новый модуль в записи требуемых служб.
4. Проверьте работу затронутых служб. Не выходите из системы до тех пор, пока не проверите возможность входа в систему.

Изменение файла the /etc/pam.conf

Прежде, чем изменять файл /etc/pam.conf, необходимо учесть некоторые факторы.

При изменении файла конфигурации `/etc/pam.conf` необходимо помнить о следующих особенностях:

- Файл всегда должен принадлежать пользователю root и группе security. Следует установить режим доступа к файлу 644, чтобы предоставить право чтения файла всем пользователям, а право вносить изменения - только пользователю root.
- Для усиления защиты рекомендуется явно настроить каждую службу с поддержкой PAM, а затем задать модуль `pam_prohibit` для всех прочих служб (ключевое слово OTHER).
- Прочитайте документацию по выбранным модулям и определите, какие управляющие флаги и опции поддерживаются модулями, и как они работают.
- Тщательно выберите порядок модулей и управляющие флаги, помня об алгоритме обработки управляющих флагов required, requisite, sufficient и optional.

Примечание: Неправильная настройка PAM может привести к тому, что вход в систему будет невозможен, поскольку параметры настройки применяются ко всем пользователям, включая пользователя root. После внесения изменений в файл обязательно тестируйте работу затронутых приложений и служб перед выходом из системы. Для восстановления системы, вход в которую невозможен, необходимо загрузить систему в режиме обслуживания и исправить файл `/etc/pam.conf`

Включение отладки PAM

Во время работы библиотека PAM может предоставлять отладочную информацию. После включения сбора отладочной информации эту информацию можно использоваться для отслеживания вызовов PAM и выявления ошибок в текущей конфигурации PAM.

Для включения отладки PAM выполните следующие действия:

1. Создайте пустой файл с именем `pam_debug` в каталоге `/etc/pam_debug` с помощью команды **touch**, если файл еще не существует. Библиотека PAM проверяет существование файла `/etc/pam_debug` и, если он существует, включает вывод syslog.
2. Измените файл `/etc/syslog.conf`, указав файл, в который будут записываться сообщения `auth` системного протокола с требуемым приоритетом. Например, для отправки сообщений уровня отладки PAM в файл `/var/log/auth.log` добавьте следующий текст в новой строке файла syslog.conf:

```
*.debug /var/log/auth.log
```

3. Создайте файл вывода, указанный в действии 2 на стр. 205, `/var/log/auth.log`, с помощью команды **touch**, если он еще не существует.
4. Для перезапуска демона syslogd с целью применения изменений выполните следующие действия:
   a. Остановите демон syslog, введя следующую команду:
      ```
      stopsrc -s syslogd
      ```
   b. Запустите демон syslog, введя следующую команду:
      ```
      startsrc -s syslogd
      ```

При следующем запуске приложения PAM отладочные сообщения будут направляться в файл вывода, заданный в файле конфигурации `/etc/syslog.conf`

OpenSSH и поддержка Kerberos версии 5

Kerberos представляет собой механизм идентификации, обеспечивающий надежную идентификацию сетевых пользователей. Применение средств шифрования данных, передаваемых между клиентами и серверами, позволяет избежать передачи текстовых паролей по сети. Кроме того, в Kerberos предусмотрена система идентификации с помощью административных маркеров и одноразовых разрешений.

Для идентификации пользователя с помощью Kerberos пользователь должен запустить команду **kinit**, позволяющую получить первоначальное одноразовое разрешение от центрального сервера Kerberos, называемого также KDC (Key Distribution Center - Центр рассылки ключей). KDC проверяет пользователя и возвращает ему первоначальное разрешение, называющееся также TGT (Ticket-Granting Ticket - начальный паспорт). После этого пользователь может запустить удаленный сеанс входа в систему, воспользовавшись Telnet с поддержкой Kerberos или OpenSSH. При этом Kerberos обеспечит идентификацию пользователя, получив от KDC его разрешения. Kerberos выполняет такую идентификацию без взаимодействия с пользователем, поэтому пользователю не нужно указывать пароль. Версия Kerberos, реализованная IBM, называется службой сетевой идентификации (NAS). NAS можно устанавливать с компакт-дисков пакета расширения AIX. Она находится в пакетах `krb5.client.rte` и `krb5.server.rte`. Начиная с июля 2003 в выпуске OpenSSH 3.6 поддерживается идентификация с помощью Kerberos 5 и проверка прав доступа с помощью NAS 1.3.

В OpenSSH версий 3.8 и выше поддерживается идентификация и проверка прав доступа Kerberos 5 с помощью NAS версии 1.4. Перед обновлением OpenSSH необходимо обновить старую версию NAS (Kerberos). OpenSSH версий 3.8.x работает только с NAS версии 1.4 или выше.

В AIX применяется OpenSSH с идентификацией Kerberos в качестве необязательного метода. Если библиотеки Kerberos в системе не установлены, то OpenSSH пропустит идентификацию Kerberos и попытается воспользоваться следующим настроенным методом идентификации (например, идентификацией AIX).

После установки Kerberos и перед началом настройки серверов Kerberos рекомендуется ознакомиться с документацией. Дополнительная информация об установке и администрировании Kerberos приведена в книге *IBM Network Authentication Service Version 1.3 for AIX : Administrator's and User's Guide*, расположенной в каталоге `/usr/lpp/krb5/doc/html/`*язык*`/ADMINGD.htm`

Информация, связанная с данной:

OpenSSH

Образы OpenSSH

Ниже рассмотрены основные этапы процесса установки образов OpenSSH:

1. Откройте веб-сайт AIX Web Download Pack Programs.

   Примечание: Образ OpenSSH входит в состав базового носителя AIX, но не устанавливается по умолчанию.

2. В разделе Дополнительная информация выберите **Загрузка**.
3. Для доступа к имеющимся пакетам введите свой ИД и пароль.
4. Выберите **OpenSSH** и нажмите кнопку **Продолжить**.
5. Для загрузки пакета примите условия лицензионного соглашения.
6. Извлеките пакет образа с помощью команды **uncompress** *имя-пакета*. Например:
   ```
   uncompress OpenSSH_6.0.0.6203.tar.Z
   ```
7. Разархивируйте пакет с помощью команды **tar -xvf** *имя-пакета*. Например:
   ```
   tar -xvf OpenSSH_6.0.0.6203.tar
   ```
8. Выполните команду **inutoc**.
9. Выполните коменду **smitty install**.
10. Выберите **Установить и обновить программное обеспечение**.
11. Выберите **Обновить установленное программное обеспечение до последнего уровня (Обновить все)**.
12. Введите точку (**.**) в поле **Устройство или каталог с программным обеспечением** и нажмите Enter.
13. Перейдите к полю **Принять новые лицензионные соглашения** и клавишей **Tab** измените значение этого поля на **Да**.
14. Дважды нажмите клавишу Enter, чтобы начать установку.

Образы OpenSSH - это образы базового уровня, а не пакеты PTF. В результате установки код предыдущей версии заменяется образами новой версии.

Настройка компиляции OpenSSH

В этом разделе описана компиляция кода OpenSSH для AIX.

При настройке OpenSSH для AIX версии 6.1 отображается примерно следующая информация:

```
OpenSSH настроена со следующими опциями:
                     Пользовательские исполняемые файлы: /usr/bin
                   Системные исполняемые файлы: /usr/sbin
               Файлы конфигурации: /etc/ssh
                   Программа Askpass: /usr/sbin/ssh-askpass
                      Страницы man: /usr/man
                          Файл PID: /etc/ssh
  Каталог chroot разделения прав доступа: /var/empty
            PATH по умолчанию для пользователя sshd: /usr/bin:/bin:/usr/sbin:/sbin:/usr/
                                  local/bin

                    Формат страниц man: man
                       Поддержка PAM: да
                   Поддержка OSF SIA: нет
                 Поддержка KerberosV: да
                 Поддержка Smartcard: нет
                   Поддержка SELinux: нет
                     Поддержка S/KEY: нет
              Поддержка оболочек TCP: да
              Поддержка пароля MD5: нет
                   Поддержка libedit: нет
  Поддержка контрактов процессов Solaris: нет
           Поддержка проектов Solaris: нет
       Хэк IP-адреса $DISPLAY: нет
           Хэк преобразования v4 в v6: нет
                 Поддержка идентификации BSD: нет
              Источник случайных чисел: OpenSSL ТОЛЬКО внутренний

              Хост: powerpc-ibm-aix6.1.0.0
          Компилятор: cc
    Флаги компилятора: -bloadmap:file -qnostdinc -qnolm -qlist -qsource -qattr=full
    Флаги препроцессора: -I/gsa/ausgsa/projects/o/openssh/freeware5/openssl-0.9.8r/
                include -I/gsa/ausgsa/projects/o/openssh/zlib -I/usr/include

          Флаги компоновщика: -L/gsa/ausgsa/projects/o/openssh/freeware5/
```

```
        lib -L/gsa/ausgsa/projects/o/openssh/zlib -L/usr/include
        -Wl,-blibpath:/usr/lib:/lib
Библиотеки: -lcrypto -lz -lc  -lcrypt -lefs   -lwrap -lpam -ldl
```

Примечание: Опция компиляции для AIX версии 6.1 и AIX версии 7.1 аналогичны, так как двоичный файл для обеих версий совпадает.

Применение OpenSSH с Kerberos

В этом разделе описана первоначальная настройка, необходимая для использования OpenSSH с Kerberos.

В этом разделе приведена информация о первоначальной настройке, необходимой для применения OpenSSH с Kerberos:

1. На серверах и клиентах OpenSSH должен существовать файл `/etc/krb5.conf`. Этот файл указывает Kerberos, какой KDC должен применяться, какова продолжительность существования каждого паспорта и т.д. Пример файла `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. Кроме того, в файл `/etc/services` на каждой клиентской системе необходимо добавить следующие службы Kerberos:

   ```
   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. Если KDC в качестве реестра пользовательской информации применяет LDAP, то рекомендуется ознакомиться с разделом “Загружаемый модуль идентификации LDAP” на стр. 156 и с документацией по Kerberos. Кроме того, необходимо проверить следующие условия:
   - На KDC работает клиент LDAP. Демона клиента LDAP можно запустить с помощью команды **secldapclntd**.
   - На сервере LDAP работает демон сервера LDAP `slapd`.

4. На сервере OpenSSH измените файл `/etc/ssh/sshd_config`, добавив в него следующие строки:

   ```
   KerberosAuthentication yes
   KerberosTicketCleanup yes
   GSSAPIAuthentication yes
   GSSAPICleanupCredentials yes
   UseDNS yes
   ```

   Если для параметра UseDNS указано значение **Yes**, то сервер ssh определяет имя подключаемого клиента посредством обратного поиска на хосте. Это необходимо, если применяется идентификация на хосте, либо для отображения в сведениях о последнем входе в систему имен хостов вместо IP-адресов.

Примечание: Иногда при выполнении обратного поиска сеансы SSH зависают из-за отсутствия доступа к серверам DNS. В этом случае, можно пропустить преобразование имен DNS в IP-адреса, установив UseDNS в значение `no`. Если параметр UseDNS отсутствует в файле `/etc/ssh/sshd_config`, используется значение по умолчанию — `UseDNS yes`.

5. На сервере SSH введите команду **startsrc -g ssh**, запускающую демона сервера SSH.
6. В клиентской системе SSH введите команду **kinit** для получения начального паспорта (TGT). Проверить получение TGT можно с помощью команды **klist**. При этом будут показаны все принадлежащие вам паспорта.
7. Подключитесь к серверу с помощью команды **ssh** *пользователь@сервер*.
8. При правильной настройке Kerberos приглашение для ввода команды показано не будет и пользователь автоматически войдет в систему сервера SSH.

Защита сети

В этих разделах описывается установка и настройка защиты IP; рассказывается о выборе нужных сетевых служб; описаны способы контроля защиты сети.

Защита TCP/IP

Протоколы TCP/IP и NFS позволяют взаимодействовать системам в сети.

В этом руководстве описаны не принципы работы TCP/IP, а предусмотренные в нем функции защиты данных. Информация об установке и начальной настройке TCP/IP приведена в разделе Протокол TCP/IP книги in *Управление сетями и средствами связи*.

В различных ситуациях требуются разные уровни защиты. Например, корпоративные правила могут требовать одного уровня защиты. Если же система имеет доступ к правительственным компьютерам, то она должна соответствовать другим требованиям к защите. Эти требования могут относиться к сети, операционной системе, приложениям и даже программам, написанным системным администратором.

В этом разделе описаны функции защиты TCP/IP в стандартном и защищенном режимах, а также различные аспекты защиты сети.

После установки TCP/IP и NFS настройте систему с помощью команды SMIT **tcpip**.

Дополнительная информация о команде **dacinet** приведена в книге *Справочник по командам*.

Защита операционной системы

Многие функции защиты TCP/IP (например, управление доступом к сети и контроль сети) зависят от операционной системы.

В следующих разделах рассмотрена защита TCP/IP.

Управление доступом к сети:

Стратегия защиты сети является продолжением стратегии защиты операционной системы и состоит из идентификации пользователей, идентификации соединения и защиты данных.

Основные компоненты следующие:

- Идентификация пользователей позволяет удаленному хосту узнать имя пользователя и пароль точно так же, как и в случае обычного входа в систему. К защищенным командам TCP/IP, таким как **ftp**, **rexec** и **telnet**, предъявляются те же требования и они проверяются с помощью того же процесса, что и защищенные команды операционной системы.
- Идентификация соединения позволяет убедиться, IP-адрес и имя хоста удаленного хоста совпадают с ожидаемыми значениями. Это гарантирует, что удаленный хост не сможет выдать себя за другой.

- Защита при экспорте и импорте данных позволяет перемещать данные между сетевыми адаптерами, не снижая уровень защиты. Например, сверхсекретные данные могут передаваться только между адаптерами, имеющими максимальный уровень защиты.

Контроль сети:

Протокол TCP/IP предусматривает контроль за сетью. С помощью подсистемы контроля он отслеживает прикладные программы.

Цель такого контроля - запись информации обо всех подозрительных действиях, которые могут привести к нарушению защиты, и именах пользователей, выполнивших такие действия.

Отслеживаются следующие события приложений:
- Доступ к сети
- Соединение
- Экспорт данных
- Импорт данных

Создание и удаление объектов отслеживаются операционной системой. Для того чтобы исключить отслеживание ненужных системных ресурсов, воспользуйтесь записями контроля за приложениями.

Защищенный путь, защищенная оболочка и защищенная клавиша внимания:

Защищенный путь применяется операционной системой, чтобы исключить нежелательное чтение данных с пользовательского терминала. Этот путь применяется вместе с защищенным соединением, например, при смене паролей или при входе в систему.

Кроме того, в операционной системе есть *защищенная оболочка* (**tsh**), выполняющая только защищенные программы, которые были протестированы и признаны надежными. TCP/IP поддерживает обе эти функции, а также *защищенную клавишу внимания* (SAK), необходимую для установления защищенных соединений между пользователем и системой. При работе с TCP/IP всегда доступна локальная SAK. Удаленная SAK предоставляется командой **telnet**.

Локальная клавиша SAK выполняет для **telnet** ту же функцию, что и для других программ операционной системы: завершает процесс **telnet** и все остальные процессы, связанные с терминалом, на котором была вызвана команда **telnet**. Кроме того, программа telnet позволяет отправить запрос удаленной системе по защищенному пути с помощью команды **telnet send sak** (в командном режиме **telnet**). Команда **telnet set sak** позволяет определить одну клавишу, которая будет отвечать за отправку запроса SAK.

Дополнительная информация о защищенной компьютерной базе приведена в разделе “Защищенная компьютерная база” на стр. 1.

Защита команд TCP/IP

Некоторые команды TCP/IP во время своего выполнения создают защищенную среду. Это команды **ftp**, **rexec** и **telnet**.

Функция **ftp** обеспечивает защиту данных при их передаче. Команда **rexec** обеспечивает защищенную среду выполнения команд на удаленном хосте. Команда **telnet** позволяет выполнять защищенный вход в систему удаленного хоста.

Команды **ftp**, **rexec** и **telnet** обеспечивают защиту только во время своего выполнения. Это означает, что они не создают защищенную среду для других команд. Для обеспечения защиты во время выполнения других операций предназначена команда **securetcpip**. Эта команда обеспечивает защиту системы путем отключения ненадежных демонов и приложений; кроме того, она предоставляет возможность защитить сетевой протокол уровня IP.

Команды **ftp**, **rexec**, **securetcpip** и **telnet** обеспечивают следующие типы защиты данных:

ftp Команда **ftp** создает защищенную среду для передачи файлов. Когда пользователь вызывает команду **ftp** для удаленного хоста, пользователю предлагается ввести свое имя и пароль. По умолчанию в качестве имени пользователя будет показано текущее имя входа в систему. У пользователя будет запрошен пароль для входа на удаленный хост.

Процесс автоматического входа в систему ищет имя пользователя и пароль удаленного хоста в локальном файле `$HOME/.netrc`. Для обеспечения защиты права доступа к файлу `$HOME/.netrc` должны быть равны 600 (чтение и запись разрешены только пользователю). В противном случае, автоматический вход в систему выполнен не будет.

Примечание: При использовании файла `.netrc` пароли хранятся в незашифрованном виде, поэтому функция автоматического входа в систему команды **ftp** будет недоступна при работе с командной **securetcpip**. Для включения этой функции удалите команду **ftp** из раздела tcpip в файле `/etc/security/config`.

При передаче файлов команда **ftp** использует два соединения TCP/IP: одно для протокола передачи файлов (FTP), а другое для передачи данных. Соединение протокола является основным и защищено, поскольку основано на надежных портах связи. Второе соединение применяется для непосредственной передачи данных, и обе стороны должны убедиться, что оно установлено с тем же хостом, что и основное. Если соединения установлены с разными хостами, то команда **ftp** выдает сообщение об ошибке и завершает работу. Это позволяет исключить перехват данных посторонним хостом.

rexec Команда **rexec** обеспечивает защищенную среду выполнения команд на удаленном хосте. Пользователь должен ввести имя и пароль.

Процесс автоматического входа в систему ищет имя пользователя и пароль удаленного хоста в локальном файле `$HOME/.netrc` с помощью команды **rexec**. Для обеспечения защиты права доступа к файлу `$HOME/.netrc` должны быть равны 600 (чтение и запись разрешены только пользователю). В противном случае, автоматический вход в систему выполнен не будет.

Примечание: При использовании файла `.netrc` пароли хранятся в незашифрованном виде, поэтому функция автоматического входа в систему команды **rexec** будет недоступна при работе с защищенной операционной системой. Для включения этой функции удалите команду **rexec** из раздела tcpip в файле `/etc/security/config`.

securetcpip
Команда **securetcpip** включает функции защиты TCP/IP. При ее вызове запрещается доступ к незащищенным командам. При вызове команды **securetcpip** будут удалены следующие команды:

- **rlogin** и **rlogind**
- **rcp**, **rsh** и **rshd**
- **tftp** и **tftpd**
- **trpt**

Команда **securetcpip** повышает уровень защиты системы. После преобразования системы команда **securetcpip** будет действовать до тех пор, пока не будет переустановлен TCP/IP.

telnet или tn
Команда **telnet** (TELNET) обеспечивает защищенную среду для входа на удаленный хост. Пользователь должен ввести имя и пароль. Терминал этого пользователя рассматривается точно так же, как и терминал, подключенный к системе напрямую. Это означает, что доступом к терминалу управляют атрибуты прав доступа. Другие пользователи не имеют доступа к терминалу, но могут отправлять на него сообщения, если у них есть права на запись. Кроме того, команда **telnet** обеспечивает доступ к защищенной оболочке удаленной команды с помощью SAK. Эта последовательность нажатия клавиш отличается от той, которая вызывает защищенный путь, и может быть определена с помощью команды **telnet**.

Вызов удаленных команд:

Пользователи хостов, перечисленных в файле /etc/hosts.equiv, могут запускать в вашей системе определенные команды, не указывая пароль.

В следующей таблице показано, как просматривать, добавлять и удалять удаленные хосты с помощью SMIT или командной строки.

Таблица 14. Задачи выполнения удаленных команд

Задача	Команда быстрого доступа SMIT	Команда или файл
Просмотр списка удаленных хостов с правами на выполнение команд	**smit lshostsequiv**	просмотрите файл /etc/hosts.equiv
Добавить удаленный хост с правами на выполнение команд	**smit mkhostsequiv**	измените файл /etc/hosts.equiv Прим.
Удалить удаленный хост с правами на выполнение команд	**smit rmhostsequiv**	измените файл /etc/hosts.equiv Прим.

Примечание: Более подробная информация о работе с этим файлом приведена в разделе "hosts.equiv File Format for TCP/IP" книги *Справочник по файлам* .

Ограничение прав пользователей FTP:

Пользователям, перечисленным в файле /etc/ftpusers, запрещен удаленный доступ по FTP. Например, допустим, что пользователю A удаленной системы известен пароль пользователя B вашей системы. Если пользователь B указан в файле /etc/ftpusers, то пользователь A не сможет от его имени передавать файлы по FTP, хотя и знает его пароль.

В следующей таблице показано, как можно просматривать, добавлять и удалять пользователей с ограниченными возможностями с помощью SMIT или командной строки.

Задачи работы с удаленными пользователями FTP

Задача	Команда SMIT	Команда или файл
Просмотр списка запрещенных пользователей FTP	**smit lsftpusers**	просмотрите файл /etc/ftpusers
Добавление запрещенного пользователя	**smit mkftpusers**	измените файл /etc/ftpusers file Прим.
Удаление запрещенного пользователя	**smit rmftpusers**	измените файл /etc/ftpusers file Прим.

Примечание: Дополнительная информация о работе с этим файлом приведена в разделе "ftpusers File Format for TCP/IP" книги *Справочник по файлам* .

Защищенные процессы

Защищенная программа (или процесс) - это сценарий оболочки, демон или программа, удовлетворяющие определенным требованиям к защите. Эти требования устанавливаются и поддерживаются стандартами Министерства обороны США, которое также сертифицирует некоторые программы.

Защищенные программы защищены на нескольких уровнях. Это уровни A1, B1, B2, B3, C1, C2 и D, где A1 - это самый высокий уровень защиты. Каждый уровень должен удовлетворять определенным требованиям. Например, уровень C2 соответствует следующим стандартам:

целостность программы
Гарантирует, что процесс работает в точности так, как предполагалось.

модульность
Означает, что исходный код разбит на модули, которые не могут напрямую влиять друг на друга.

принцип наименьших привилегий
Утверждает, что пользователь должен обладать минимальными возможными привилегиями. Например, если пользователь должен только просматривать файл, то у него не должно быть прав на изменение файла.

запрет на повторное использование объектов
Исключает возможность, например, случайного обнаружения пользователем сегмента памяти с конфиденциальными данными, который был помечен для удаления, но еще не был очищен.

TCP/IP содержит несколько защищенных и множество незащищенных демонов.

Примеры защищенных демонов:
- **ftpd**
- **rexecd**
- **telnetd**

Примеры незащищенных демонов:
- **rshd**
- **rlogind**
- **tftpd**

Отдельная защищенная система должна работать с защищенной компьютерной базой, то есть должна быть защищена. В сети должны быть защищены все файловые серверы, шлюзы и другие хосты.

Сетевая защищенная компьютерная база

Сетевая защищенная компьютерная база (NTCB) состоит из программного и аппаратного обеспечения, гарантирующего защиту данных в сети. В этом разделе описаны компоненты NTCB, относящиеся к TCP/IP.

Функции аппаратной защиты выполняются сетевыми адаптерами, используемыми TCP/IP. Эти адаптеры принимают только данные, отправленные локальной системе, и пересылают все остальные данные.

Программная часть NTCB состоит из набора защищенных программ. Программы и файлы, входящие в состав защищенной системы, перечислены по каталогам в следующих таблицах:

Каталог /etc

Имя	Владелец	Группа	Режим	Права доступа
gated.conf	root	system	0664	rw-rw-r—
gateways	root	system	0664	rw-rw-r—
hosts	root	system	0664	rw-rw-r—
hosts.equiv	root	system	0664	rw-rw-r—
inetd.conf	root	system	0644	rw-r—r—
named.conf	root	system	0644	rw-r—r—
named.data	root	system	0664	rw-rw-r—
networks	root	system	0664	rw-rw-r—
protocols	root	system	0644	rw-r—r—
rc.tcpip	root	system	0774	rwxrwxr—
resolv.conf	root	system	0644	rw-rw-r—
services	root	system	0644	rw-r—r—
3270.keys	root	system	0664	rw-rw-r—
3270keys.rt	root	system	0664	rw-rw-r—

Каталог /usr/bin

Имя	Владелец	Группа	Режим	Права доступа
host	root	system	4555	r-sr-xr-x
hostid	bin	bin	0555	r-xr-xr-x
hostname	bin	bin	0555	r-xr-xr-x
finger	root	system	0755	rwxr-xr-x
ftp	root	system	4555	r-sr-xr-x
netstat	root	bin	4555	r-sr-xr-x
rexec	root	bin	4555	r-sr-xr-x
ruptime	root	system	4555	r-sr-xr-x
rwho	root	system	4555	r-sr-xr-x
talk	bin	bin	0555	r-xr-xr-x
telnet	root	system	4555	r-sr-xr-x

Каталог /usr/sbin

Имя	Владелец	Группа	Режим	Права доступа
arp	root	system	4555	r-sr-xr-x
fingerd	root	system	0554	r-xr-xr—
ftpd	root	system	4554	r-sr-xr—
gated	root	system	4554	r-sr-xr—
ifconfig	bin	bin	0555	r-xr-xr-x
inetd	root	system	4554	r-sr-xr—
named	root	system	4554	r-sr-x—
ping	root	system	4555	r-sr-xr-x
rexecd	root	system	4554	r-sr-xr—
route	root	system	4554	r-sr-xr—
routed	root	system	0554	r-xr-x—-
rwhod	root	system	4554	r-sr-xr—
securetcpip	root	system	0554	r-xr-xr—
setclock	root	system	4555	r-sr-xr-x
syslogd	root	system	0554	r-xr-xr—
talkd	root	system	4554	r-sr-xr—
telnetd	root	system	4554	r-sr-xr—

Каталог /usr/ucb

Имя	Владелец	Группа	Режим	Права доступа
tn	root	system	4555	r-sr-xr-x

Каталог /var/spool/rwho

Имя	Владелец	Группа	Режим	Права доступа
rwho (каталог)	root	system	0755	drwxr-xr-x

Защита данных и информации

Функция защиты TCP/IP не шифрует передаваемые по сети пользовательские данные.

Поэтому следует помнить о риске, связанный с возможностью перехвата передаваемых по сети конфиденциальных данных (например, паролей), и принимать соответствующие контрмеры.

При использовании TCP/IP в среде Министерства обороны США (DOD) в некоторых случаях необходимо следовать инструкциям DOD 5200.5 и NCSD-11, относящимся к защите соединений.

Управление доступом к порту TCP на уровне пользователей и самостоятельный контроль доступа к портам Internet

Функция самостоятельного контроля доступа к портам Internet (DACinet) обеспечивает контроль доступа пользователей к портам TCP на хостах AIX.

В AIX для передачи информации о пользователе и группе может использоваться дополнительный заголовок TCP. Функция DACinet позволяет администраторам удаленной системы управлять доступом к своей системе на основе целевого порта, исходного ИД пользователя и хоста.

Кроме того, с помощью функции DACinet можно запретить доступ к локальным портам всем пользователям, кроме root. В системах UNIX, к числу которых относится и AIX, порты с номерами меньше 1024 считаются привилегированными и могут открываться только пользователем root. Система AIX позволяет точно так же настроить и порты с номерами выше 1024 (доступ только пользователя root).

Вы можете разрешить или запретить системам без поддержки DACinet обращаться к системе с поддержкой DACinet. По умолчанию такой доступ запрещен. Включив поддержку DACinet, вы не сможете в дальнейшем ее выключить.

В качестве параметров команды **dacinet** можно указывать имена хостов, IP-адреса в десятичном формате с точками и адреса сетей с указанной длиной префикса.

В следующем примере указан один хост с именем *host.domain.org*:

```
host.domain.org
```

В следующем примере указан один хост с IP-адресом 10.0.0.1:

```
10.0.0.1
```

В следующем примере указана сеть, первые 24 разряда которой (длина префикса сети) равны 10.0.0.0:

```
10.0.0.0/24
```

В эту сеть входят адреса от 10.0.0.1 до 10.0.0.254.

Управление доступом к службам TCP:

В DACinet используется файл запуска `/etc/rc.dacinet` и файлы конфигурации `/etc/security/priv`, `/etc/security/services` и `/etc/security/acl`.

В файле `/etc/security/services` перечислены порты, исключаемые при проверке ACL. Формат этого файла такой же, как у файла `/etc/services`. Для создания этого файла рекомендуется скопировать образец из каталога `/etc` в `/etc/security` и удалить номера портов, для которых должна выполняться проверка ACL. Списки ACL хранятся в двух местах. Текущие ACL расположены в ядре; их можно прочесть с помощью

команды dacinet aclls. ACL, которые будут активизированы при следующей загрузке /etc/rc.tcpip, хранятся в /etc/security/acl. Применяется следующий формат:

```
служба хост/длина_префикса [пользователь|группа]
```

где служба указывается в цифровом виде или в формате /etc/services, хост - по имени или адресу сети с маской подсети, а имя пользователя и группа - с помощью префиксов u: or g:. Если имя пользователя и группа не указаны, то ACL учитывает только отправляющий хост. Для явного запрета доступа укажите перед именем службы символ -. ACL просматриваются до первого совпадения. Следовательно, если вы хотите разрешить доступ всей группе, за исключением одного пользователя, то укажите правило для этого пользователя до правила для группы.

В файле /etc/services есть две записи с номерами портов, не поддерживаемыми в AIX. Перед вызовом команды **mkCCadmin** системный администратор должен удалить эти записи. Удалите из файла /etc/services следующие записи:

```
sco_printer     70000/tcp     sco_spooler    # For System V print IPC
sco_s5_port     70001/tcp     lpNet_s5_port  # For future use
```

Примеры использования DACinet:

Например, если с помощью DACinet разрешить доступ к порту TCP/25 для входящих соединений только пользователю root с поддержкой DACinet, то к этому порту смогут обращаться только пользователи root с хостов AIX. Таким образом вы сможете оградить себя от возможных попыток других пользователей отправлять от вашего имени электронную почту путем подключения к порту TCP/25.

В следующем примере доступ к протоколу X11 разрешается только пользователю root. Убедитесь, что запись для X11 удалена из файла /etc/security/services, так что ACL будут применяться для этой службы.

Если все подключенные системы находятся в подсети 10.1.1.0/24, то для того, чтобы разрешить доступ к X (TCP/6000) только пользователю root, необходимо указать в файле /etc/security/acl следующую запись ACL:

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

Для того чтобы разрешить доступ по Telnet только пользователям в группе friends из любой системы, после удаления записи для telnet из /etc/security/services введите следующую запись ACL:

```
telnet    0.0.0.0/0   g:friends
```

Следующие записи позволяют запретить доступ к Web-серверу пользователю fred, но разрешить такой доступ всем остальным пользователям:

```
-80    0.0.0.0/0 u:fred
80     0.0.0.0/0
```

Привилегированные порты для локальных служб:

Для того чтобы запретить выделение серверам конкретных портов, сделайте эти порты привилегированными.

Обычно всем пользователям разрешается открывать любые порты с номерами выше 1024. Например, можно выделить Web-серверу порт 8080, а серверу SOCKS - порт 1080. Для создания привилегированных портов в работающей системе служит команда **dacinet setpriv**. В файле /etc/security/priv следует перечислить привилегированные порты, применяемые при запуске системы.

Порты можно указать либо с помощью символьного имени, как в файле /etc/services, либо по номеру. Следующие записи запрещают всем пользователям, кроме root, выделять серверам SOCKS и Lotus Notes их обычные порты:

```
1080
lotusnote
```

Примечание: Эта функция не запрещает пользователям запускать программы. Она лишь запрещает пользователям выделять запускаемым службам конкретные порты.

Сетевые службы

В этом разделе приведена информация, которая поможет вам идентифицировать сетевые службы с открытыми портами связи, а также обеспечить их защиту.

Формат портов

В следующей таблице описан формат известных портов в AIX.

Примечание: Данный список был создан на основе сведений о большом количестве систем AIX с различными настройками и различным программным обеспечением.

В следующий список могли войти не все форматы портов для программного обеспечения к AIX:

Порт/Протокол	Служебное имя	Псевдонимы
13/tcp	daytime	Daytime (RFC 867)
13/udp	daytime	Daytime (RFC 867)
21/tcp	ftp	File Transfer [Control]
21/udp	ftp	File Transfer [Control]
23/udp	telnet	Telnet
23/udp	telnet	Telnet
25/tcp	smtp	Simple Mail Transfer
25/udp	smtp	Simple Mail Transfer
37/tcp	time	Time
37/udp	time	Time
111/tcp	sunrpc	SUN Remote Procedure Call
111/udp	sunrpc	SUN Remote Procedure Call
161/tcp	snmp	SNMP
161/udp	snmp	SNMP
199/tcp	smux	SMUX
199/udp	smux	SMUX
512/tcp	exec	выполнение удаленного процесса;
513/tcp	login	удаленный вход в систему, как 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

Порт/Протокол	Служебное имя	Псевдонимы
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

Определение системных служб с открытыми портами связи

Приложения клиент-сервер открывают на сервере порты связи, обеспечивая прием поступающих от клиентов запросов.

Поскольку открытые порты связи представляют собой потенциальную угрозу безопасности системы, то необходимо определить, какие приложения имеют открытые порты связи, и, если эти порты не нужны, закрыть их. Такой подход позволяет вам определить, какие системы доступны любому пользователю, имеющему доступ к Internet.

Для определения открытых портов выполните следующие действия:

1. Определите список служб с помощью следующей команды **netstat**:

   ```
   # netstat -af inet
   ```

   Ниже приведен пример вывода этой команды. В последней столбце вывода команды **netstat** указано состояние каждой службы. Службы, ожидающие установления соединений, находятся в состоянии ОЖИДАНИЕ (LISTEN).

Пример вывода при выполнении команды **netstat**.

Активное соединение с Internet (включая серверы)

Протокол	Recv-Q	Send-Q	Локальный адрес	Удаленный адрес	(состояние)
tcp4	0	0	*.echo	*.*	ОЖИДАНИЕ
tcp4	0	0	*.discard	*.*	ОЖИДАНИЕ
tcp4	0	0	*.daytime	*.*	ОЖИДАНИЕ
tcp	0	0	*.chargen	*.*	ОЖИДАНИЕ
tcp	0	0	*.ftp	*.*	ОЖИДАНИЕ
tcp4	0	0	*.telnet	*.*	ОЖИДАНИЕ
tcp4	0	0	*.smtp	*.*	ОЖИДАНИЕ
tcp4	0	0	*.time	*.*	ОЖИДАНИЕ
tcp4	0	0	*.www	*.*	ОЖИДАНИЕ

Пример вывода при выполнении команды **netstat**.

Активное соединение с Internet (включая серверы)

Протокол	Recv-Q	Send-Q	Локальный адрес	Удаленный адрес	(состояние)
tcp4	0	0	*.sunrpc	*.*	ОЖИДАНИЕ
tcp	0	0	*.smux	*.*	ОЖИДАНИЕ
tcp	0	0	*.exec	*.*	ОЖИДАНИЕ
tcp	0	0	*.login	*.*	ОЖИДАНИЕ
tcp4	0	0	*.shell	*.*	ОЖИДАНИЕ
tcp4	0	0	*.klogin	*.*	ОЖИДАНИЕ
udp4	0	0	*.kshell	*.*	ОЖИДАНИЕ
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. Откройте файл /etc/services и проверьте, каким номерам портов соответствуют службы Internet Assigned Numbers Authority (IANA).

   Ниже приведен фрагмент файла /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. Закройте ненужные порты, удалив работающие службы.

Примечание: Порт 657 используется подсистемой Контроля и управления ресурсами (RMC) для связи между узлами. Этот порт нельзя блокировать, или каким-либо иным способом ограничивать его использование.

Идентификация сокетов TCP и UDP

Для идентификации сокетов TCP, находящихся в состоянии ОЖИДАНИЕ, и простаивающих сокетов UDP, ожидающих поступления данных, служит команда **lsof**, являющаяся разновидностью команды **netstat -af**.

Например, для просмотра списка сокетов TCP, находящихся в состоянии ОЖИДАНИЕ, а также сокетов UDP, находящихся в состоянии ПРОСТАИВАЕТ, введите следующую команду **lsof**:

```
# lsof -i | egrep "КОМАНДА|ОЖИД|UDP"
```

Вывод будет выглядеть примерно следующим образом:

Команда	PID	Имя	FD	Тип	Устр.	Разм.	Узел	Имя
dtlogin	2122	root	5u	IPv4	0x70053c00	0t0	UDP	*:xdmcp
dtlogin	2122	root	6u	IPv4	0x70054adc	0t0	TCP	*:32768(ОЖИД.)
syslogd	2730	root	4u	IPv4	0x70053600	0t0	UDP	*:syslog
X	2880	root	6u	IPv4	0x70054adc	0t0	TCP	*:32768(ОЖИД.)
X	2880	root	8u	IPv4	0x700546dc	0t0	TCP	*:6000(ОЖИД.)
dtlogin	3882	root	6u	IPv4	0x70054adc	0t0	TCP	*:32768(ОЖИД.)
dtgreet	4656	root	6u	IPv4	0x70054adc	0t0	TCP	*:32768(ОЖИД.)

После определения ИД процесса (PID) вы можете с помощью следующей команды получить дополнительную информацию о программе:

```
" # ps -fp PID#"
```

Вывод содержит полное имя программы, по которому вы можете найти эту программу.

Защита протокола IP

Функция Защита IP-пакетов обеспечивает защиту данных, передаваемых на уровне IP через Internet и по внутренним сетям компании.

Защита IP - Обзор

Защита IP позволяет отдельным пользователям и организациям защитить данные всех приложений, не изменяя эти приложения. Таким образом можно защитить любые передаваемые данные от электронной почты до собственных приложений компании.

Защита IP и операционная система:

В состав операционной системы входят средства защиты IP (IPsec), в которых применяются стандартные открытые технологии защиты, разработанные группой Internet Engineering Task Force (IETF).

Модуль IPsec обеспечивает защиту данных на уровне IP стека связи посредством шифрования. Эта процедура прозрачна для приложений и не требует их изменения. Среда IPsec выбрана группой IETF в качестве стандарта для защиты сетей как протокола IPv4, так и IPv6.

В среде IPsec применяются следующие средства защиты данных:

Идентификация
Процедура, позволяющая проверить подлинность отправителя данных.

Проверка целостности

Процедура, позволяющая убедиться в том, что данные не подвергались изменениям в ходе передачи по сети.

Шифрование

Процедура обеспечения секретности при передаче данных, использующая технологии скрытия данных и частных IP-адресов при передаче по открытой сети.

Применяемые алгоритмы идентификации позволяют убедиться в подлинности отправителя данных и в том, что данные не изменялись в ходе передачи. Для этого каждый отправляемый пакет (включая фиксированные заголовки IP) обрабатывается с помощью специальной хэш-функции шифрования, использующей секретный ключ. Получатель дешифрует данные по тому же алгоритму. Если в ходе дешифрования выяснится, что данные подвергались изменению или были зашифрованы неправильным ключом, дейтаграмма отбрасывается.

Шифрование заключается в преобразовании данных по специальному алгоритму с применением ключа. На выходе алгоритм выдает *зашифрованный текст*. Шифрование нужно для того, чтобы данные нельзя было интерпретировать, если они будут перехвачены при передаче по сети. После получения зашифрованного текста получатель преобразует его в исходные данные с помощью того же алгоритма и, в случае симметричного шифрования, того же ключа. В процессе дешифрования происходит проверка целостности зашифрованных данных.

Для реализации этих функций в системе IPsec применяются протоколы ESP (Encapsulating Security Payload) и AH (Authentication Header). Протокол ESP применяется для шифрования пакетов IP. Он добавляет к пакету заголовок ESP и помещает зашифрованные данные в тело пакета ESP.

Протокол AH может применяться как совместно с протоколом ESP, так и независимо от него для проверки целостности данных. Протокол AH применяется для шифрования статических полей заголовка IP и добавления уникального контрольного значения к пакету, полученного путем применения хэш-алгоритма к данным, содержащимся в пакете. Получатель вычисляет фактическое контрольное значение и сравнивает его с контрольным значением, указанным в пакете. Если они совпадают, пакет не изменялся при передаче по сети.

Возможности защиты IP:

В этом разделе представлена информация о возможностях защиты IP.

Internet Key Exchange для операционной системы AIX предоставляет следующие функции:

- Поддерживает AES 128-, 92- и 256-разрядные алгоритмы.
- Аппаратное ускорение передачи данных через адаптеры 10/100 Мбит Ethernet PCI Adapter II.
- Поддержка AH согласно RFC 2402 и поддержка ESP согласно RFC 2406.
- Поддержка статических туннелей для работы с системами, не поддерживающими автоматическое обновление ключей IKE, а также для протокола IPv6.
- Поддержка туннелей с инкапсуляцией данных для работы в открытых сетях.
- Алгоритмы идентификации HMAC (Hashed Message Authentication Code) MD5 (Message Digest 5) и HMAC SHA (Secure Hash Algorithm).
- Алгоритмы шифрования DES с 56-разрядным ключом, CBC с 64-разрядным начальным вектором (IV), Triple DES, DES CBC 4 (32-разрядный IV) и AES CBC.
- Поддержка двойных стеков IP (IPv4 и IPv6).
- Инкапсуляция и фильтрация потоков данных IPv4 и IPv6. Эти стеки IP работают независимо друг от друга, поэтому функцию защиты IP можно настраивать независимо для разных стеков.
- Фильтрация защищенных и незащищенных потоков данных по разнообразным признакам, например, по IP-адресам отправителей и получателей, интерфейсам, протоколам, номерам портов и т.д.
- Автоматическое создание и удаление правил фильтрации для большинства типов туннелей.

- Поддержка имен хостов (вместо IP-адресов) при определении туннелей и правил фильтрации. IP-адреса хостов определяются по их именам автоматически (при наличии службы DNS).
- Ведение протоколов событий защиты IP с помощью демона **syslog**.
- Трассировка системы и сбор статистики для устранения неполадок.
- По умолчанию пользователь самостоятельно определяет, разрешена ли пересылка посторонних данных по туннелю.

Следующие дополнительные функции доступны в Internet Key Exchange для AIX 6.1 TL 05 и более поздних версиях:
- Поддержка IPSec согласно RFC 4301, поддержка AH согласно RFC 4302 и поддержка ESP согласно RFC 4303
- Алгоритмы идентификации шифра на основе Кода идентификации сообщения (CMAC) AES XCBC
- Алгоритмы шифрования включают в себя AES 128-разрядный, 192-разрядный, 256-разрядный GCM (16-разрядный IV), AES-128-GMAC, AES-192-GMAC и AES-256-GMAC
- Поддержка диапазона портов для правил фильтрации
- Расширенные порядковые номера

Возможности протокола Internet Key Exchange (IKE):

Следующие функции применимы в Internet Key Exchange для AIX.

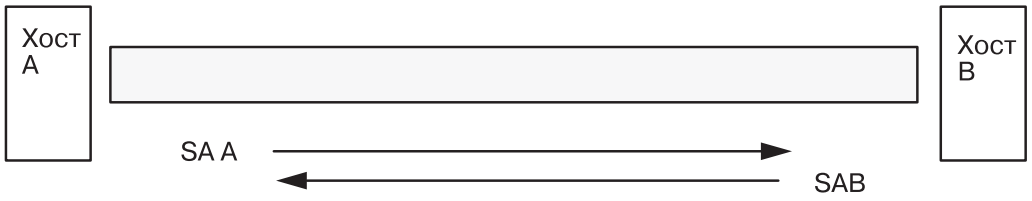
Следующие дополнительные функции доступны в Internet Key Exchange для AIX 6.1 и более поздних версиях:
- Поддержка AH для 256-разрядное хеширование HMAC SHA2 (TL 04 и выше).
- Поддержка шифрования по протоколу ESP для алгоритмов GCM AES 128-, 192- и 256-бит (16-бит IV), GMAC AES 128-, 192-, 256-бит; поддержка идентификации ESP с помощью HMAC MD5 и HMAC SHA1 (TL 04 и выше).
- Поддерживаются IKEv1 (RFC2409) и IKEv2 (RFC4306) (TL 02 и выше). IKEv1 поддерживается демоном **isakmpd** и IKEv2 поддерживается демоном **ikev2d** (TL 02 и выше). Туннели IKEv1 и IKEv2 могут сосуществовать.
- Поддержка алгоритмов целостности данных CMAC_AES_XCBC и HMAC_SHA2_256 (TL 04 и выше).
- Поддержка алгоритма PRF PRF_SHA2_256 (TL 04 и выше).
- Поддержка групп Diffie Hellman 14, 19 и 24 (TL 04 и выше).

Конфигурации защиты:

Основной элемент организации защищенной связи - это *конфигурация защиты*. Конфигурация защиты содержит параметры передачи данных по каналу связи.

В каждом направлении передачи данных и для каждого типа заголовков (AH и ESP) применяется собственная конфигурация защиты. В конфигурации защиты указываются IP-адреса отправителя и получателя данных, уникальный идентификатор SPI (индекс параметров защиты), сведения о протоколах идентификации и шифрования, ключи идентификации и шифрования и срок действия ключей. На следующем рисунке проиллюстрирована конфигурация защиты связи между хостами A и B.

На этом рисунке показан виртуальный туннель между хостами A и B. Конфигурация защиты B обозначена



SA = Конфигурация защиты, состоящая из следующих элементов:

Целевой адрес
Индекс стратегии защиты
Ключ
Алгоритм и формат шифрования
Алгоритм идентификации
Время жизни ключей

Рисунок 6. Установление защищенного туннеля между хостами A и B

стрелкой, направленной от хоста B к хосту A. Конфигурация защиты включает целевой адрес, SPI, ключ, алгоритм шифрования, а также формат, алгоритм идентификации и значение срока действия ключа.

Средства работы с ключами выполняют функцию согласования и вычисления ключей для конфигураций защиты.

Туннели и ключи:

Согласование конфигураций защиты, необходимых для настройки защищенного соединения между двумя хостами, и управление ими можно осуществить с помощью туннелей.

В зависимости от способа работы с ключами, различают следующие типы туннелей:

- Туннели IKE (с динамическим изменением ключей, стандарт IETF)
- Статические туннели (без изменения ключей, стандарт IETF)

Поддержка туннелей IKE:

Туннели IKE работают по стандартам ISAKMP/Oakley (Internet Security Association and Key Management Protocol), разработанным группой IETF. В этих стандартах реализованы процедуры защищенного согласования и обновления параметров защиты и защищенного обмена ключами.

Поддерживаются следующие типы идентификации:

- Подготовленный ключ.
- Подписи цифрового сертификата X.509v3.
- В AIX 6.1 TL 04 и более поздних версиях IKEv2 поддерживает подписи цифрового сертификата ECDSA-256 в составе метода идентификации X509v3, основанного на цифровых сертификатах.

Согласование выполняется в два этапа. На первом этапе выполняется идентификация сторон и выбираются алгоритмы обеспечения защиты второго этапа. На втором этапе выполняется согласование параметров защиты IP для передачи данных, происходят создание конфигураций защиты и ключей и обмен конфигурациями и ключами.

В следующей таблице перечислены алгоритмы идентификации, которые могут применяться с протоколами защиты AH и ESP в туннелях IKE.

Таблица 15. Алгоритмы идентификации для поддержки туннеля IKE

Алгоритм	AH IP версий 4 и 6	ESP IP версий 4 и 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

Поддержка статических туннелей:

Поддержка статических туннелей предусмотрена только для совместимости с системами, не поддерживающими динамические туннели. Недостаток статических туннелей заключается в отсутствии возможности регулярного обновления ключей. В течение всего срока существования туннеля для шифрования и идентификации применяются одни и те же ключи.

В следующей таблице перечислены алгоритмы идентификации, которые могут применяться с протоколами защиты AH и ESP в статических туннелях.

Алгоритм	AH IP версии 4	AH IP версии 6	ESP IP версии 4	ESP IP версии 6
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

Поскольку туннели IKE обеспечивают более надежную защиту, статические туннели рекомендуется применять только при отсутствии альтернативы.

Встроенные фильтры:

Фильтрация - это выборочный прием поступающих пакетов в соответствии с установленными пользователем правилами. Пользователи и администраторы системы могут контролировать потоки данных между локальным и удаленными хостами.

Фильтрация может осуществляться по различным параметрам пакетов, например, по IP-адресам отправителей и получателей, версии протокола IP (4 или 6), маскам подсетей, применяемым протоколам, номерам портов, маршрутам, фрагментации, интерфейсам или определениям туннелей.

Правила фильтрации позволяют точно указать потоки данных, которые могут передаваться по туннелю. При создании статических туннелей между индивидуальными хостами автоматически создаются правила фильтрации, направляющие все исходящие данные с локального хоста в туннель. Если требуется иная маршрутизация (например, передача по туннелю только данных, адресованных в конкретную подсеть), можно соответствующим образом изменить или заменить правила фильтрации.

При работе с туннелями IKE правила фильтрации создаются и добавляются в таблицу фильтров в момент активации туннеля.

Для упрощения настройки защиты IP при изменении и удалении туннелей правила фильтрации автоматически корректируются или удаляются. Предусмотрены средства импорта и экспорта, упрощающие перенос фильтров и определений туннелей на другие системы, что удобно при работе с большим числом компьютеров.

Правила фильтрации в первую очередь предназначены для обработки данных, передаваемых по туннелю, однако они могут применяться и для обработки прочих данных. Это предоставляет возможности по организации брандмауэра в операционной системе на случай, если вам требуется разграничить передачу данных между системами во внутренней сети, или если в вашей сети нет полноценного брандмауэра. В этом случае фильтры можно рассматривать как вторую линию защиты системы.

После создания правила фильтрации заносятся в таблицу и загружаются в ядро операционной системы. Когда пакет готов к передаче по сети, он проверяется на соответствие всем правилам фильтрации в том порядке, в котором они указаны в таблице. По результатам проверки принимается решение о том, разрешена ли отправка пакета, и нужно ли его отправлять через туннель. Правила фильтрации просматриваются до тех пор, пока не будет найдено подходящее правило или правило, применяемое по умолчанию.

С помощью средств защиты IP можно фильтровать и незащищенные пакеты, передаваемые по сетям, в которых не обязательна защита данных.

Поддержка цифровых сертификатов:

Средства защиты IP поддерживают цифровые сертификаты X.509 версии 3.

С помощью Диспетчера сертификатов можно создавать запросы на выдачу сертификатов, обслуживать базу ключей и выполнять прочие административные действия.

Общие сведения о цифровых сертификатах приведены в разделе Настройка цифровых сертификатов. Диспетчер ключей и его функции описаны в разделе Работа с программой Диспетчер ключей IBM.

Виртуальные частные сети и защита IP:

Виртуальные частные сети (VPN) предназначены для организации обмена данными между защищенными сетями по открытым сетям (например, по сети Internet).

Фактически VPN представляет собой защищенный туннель, проложенный по открытой сети между разными защищенными сетями организации (например, между разными офисами компании). Это позволяет сэкономить значительные средства, поскольку компаниям не нужно тратиться на прокладку выделенных линий между удаленными офисами, платить за междугородную связь и нести прочие издержки, которые были бы вызваны непосредственной прокладкой канала связи. Для организации виртуальной частной сети можно воспользоваться стандартом защиты IPsec, выбранным группой IETF в качестве промышленного стандарта организации защищенных сетей для протоколов IPv4 и IPv6.

Рекомендуем вам ознакомиться с главой 9 книги *A Comprehensive Guide to Virtual Private Networks, Volume III: Cross-Platform Key and Policy Management*, ISBN SG24-5309-00. В ней приведена ценная информация о проектировании и организации виртуальных частных сетей в операционной системе AIX. Это руководство опубликовано в сети Internet по адресу http://www.redbooks.ibm.com/redbooks/SG245309.html.

Установка функции защиты IP-пакетов

Функция защиты IP-пакетов в AIX устанавливается и загружается отдельно.

Необходимо установить следующие наборы файлов:

- `bos.net.ipsec.rte` (среда времени выполнения для поддержки защиты IP-пакетов в ядре и командах)
- `bos.msg.`*LANG*`.net.ipsec` (где *LANG* - указанный язык, например ru_RU)
- `bos.net.ipsec.keymgt`
- `clic.rte` (CryptoLite для C, набор файлов для криптоалгоритмов DES, тройной DES и AES).

Для поддержки цифровых подписей в IKE необходимо также установить набор файлов `gskit.rte` или `gskkm.rte` из пакета расширения.

После установки защита IP-пакетов может быть загружена отдельно для протоколов IP версий 4 и 6 либо способом, рекомендованным в разделе “Загрузка защиты IP-пакетов”, либо с помощью команды **mkdev**.

Загрузка защиты IP-пакетов:

Если защита IP-пакетов запущена, то можно автоматически загрузить ее модули с помощью SMIT. Кроме того, SMIT гарантирует загрузку расширений ядра и демонов IKE в правильном порядке.

Примечание: Загрузка защиты IP-пакетов включает фильтрацию. Перед загрузкой необходимо проверить правила фильтрации. В противном случае вся связь системы может быть заблокирована.

Если загрузка выполнена успешно, то команда **lsdev** покажет для устройств защиты IP состояние `Доступно`.

```
lsdev -C -c ipsec

   ipsec_v4 Доступно Расширение: защита IP-пакетов версии 4
   ipsec_v6 Доступно Расширение: защита IP-пакетов версии 6
```

После загрузки расширения ядра для поддержки защиты IP-пакетов можно перейти к настройке туннелей и таблиц фильтрации.

Планирование конфигурации защиты IP

Для настройки защиты IP вам в первую очередь понадобится настроить туннели и фильтры.

Если все данные передаются через простой туннель, то правила фильтрации могут быть созданы автоматически. Если требуется более сложная фильтрация, то правила фильтрации следует настраивать отдельно.

Защиту IP можно настроить с помощью модуля Виртуальные частные сети или программы SMIT. При использовании SMIT можно воспользоваться следующими командами:

smit ips4_basic
: Первоначальная настройка IP версии 4.

smit ips6_basic
: Первоначальная настройка IP версии 6.

Перед настройкой защиты IP необходимо выбрать способ защиты: например, вы можете воспользоваться туннелями и/или фильтрами, выбрать тип туннеля и т.д. В следующих разделах приведена информация, которая поможет вам принять правильное решение:

Аппаратное ускорение:

Адаптер 10/100 Mbps Ethernet PCI Adapter II (код 4962) обеспечивает стандартные средства защиты IP и позволяет снять с операционной системы AIX часть нагрузки по обеспечению защиты IP.

Если в системе AIX применяется адаптер 10/100 Mbps Ethernet PCI Adapter II, то стек защиты IP применяет следующие возможности этого адаптера:

- Шифрование и расшифровка данных с помощью алгоритмов DES и Triple DES.

- Идентификация с помощью алгоритмов MD5 и SHA-1.
- Хранение информации, связанной с защитой

Эти функции адаптера используются вместо программных алгоритмов. Адаптер 10/100 Mbps Ethernet PCI Adapter II может применяться со статическими туннелями и туннелями IKE.

Функция аппаратного ускорения защиты IP поддерживается в наборах файлов `bos.net.ipsec.rte` и `devices.pci.1410ff01.rte` 5.1.0.25 и более поздних уровней.

Существует ограничение на число функций защиты, которые могут быть переданы адаптеру на принимающей стороне. На передающей стороне все пакеты, использующие поддерживаемую конфигурацию, обрабатываются адаптером. В некоторых конфигурациях туннелей обработку пакетов нельзя поручить адаптеру.

Адаптеры 10/100 Mbps Ethernet PCI Adapter II поддерживают следующие функции:
- Шифрование DES, 3DES или NULL с помощью ESP
- Идентификация HMAC-MD5 или HMAC-SHA-1 с помощью ESP или AH, но не оба способа одновременно. (Если одновременно используется ESP и AH, то сначала должно выполняться обращение к ESP. Это правило всегда действует для туннелей IKE, но для статических туннелей пользователь может задавать порядок вручную.)
- Поддержка режима транспорта и туннеля
- Обработка пакетов IPV4

Примечание: Адаптер 10/100 Mbps Ethernet PCI Adapter II не может обрабатывать пакеты с опциями IP.

Для активации функция защиты IP адаптера 10/100 Mbps Ethernet PCI Adapter II может потребоваться отключить сетевой интерфейс и активировать функцию обработки IPsec.

Для отключения сетевого интерфейса выполните с помощью SMIT следующие действия:

Для активизации функции обработки IPsec выполните с помощью SMIT следующие действия:
1. Войдите в систему как пользователь **root**.
2. Введите в командной строке `smitty eadap` и нажмите Enter.
3. Выберите опцию **Изменить/показать параметры адаптера Ethernet** и нажмите Enter.
4. Выберите адаптер 10/100 Mbps Ethernet PCI Adapter II и нажмите Enter.
5. Укажите в поле Обработка IPsec значение да и нажмите Enter.

Для отключения сетевого интерфейса с помощью командной строки введите следующую команду:
```
# ifconfig enX detach
```

Для включения атрибута обработки IPsec с помощью командной строки введите следующую команду:
```
# chdev -l entX -a ipsec_offload=yes
```

Для проверки атрибута обработки IPsec введите следующую команду:
```
# lsattr -El entX detach
```

Для отключения атрибута обработки IPsec введите следующую команду:
```
# chdev -l entX -a ipsec_offload=no
```

С помощью команды **enstat** убедитесь, что в конфигурации туннеля используется атрибут обработки IPsec. Команда **enstat** показывает полную статистическую информацию о передаче и приеме пакетов IPsec при включенном атрибуте обработки IPsec. Например, если применяется интерфейс Ethernet **ent1**, введите следующую команду:

```
# entstat -d ent1
```

Вывод команды будет выглядеть примерно следующим образом:

```
.
.
.
Статистические данные для 10/100 Mbps Ethernet PCI Adapter II (1410ff01):
-------------------------------------------
.
.
.
Передано пакетов IPsec: 3
Отброшено переданных пакетов IPsec: 0
Получено пакетов IPsec: 2
Отброшено полученных пакетов IPsec: 0
```

Параметр сети:

В зависимости от числа туннелей в конфигурации можно увеличить максимальный размер буфера для сокета.

Если в среде запущено большое число туннелей, а для параметра **sb_max** указано значение по умолчанию, то демон IKE и демон администратора туннелей могут перестать отвечать вследствие высокой нагрузки на сеть.

Рекомендуемые значения параметра **sb_max**:

- 10 МБ для 500 туннелей
- 20 МБ для 1000 туннелей

Информация, связанная с данной:

Настройка sb_max

Туннели и фильтры:

Защита IP работает с двумя разными компонентами - *туннелями* и *фильтрами*. Туннели требуют применения фильтров, однако фильтры могут применяться и отдельно от туннелей.

Фильтрация - это функция, принимающая или отклоняющая поступающие и исходящие пакеты в зависимости от множества параметров, задаваемых *правилами*. Эта функция позволяет системному администратору настраивать средства управления передачей данных между этим и другими хостами сети. Фильтрация может осуществляться на основании самых разных свойств пакетов, включая исходный и целевой адрес, версия IP (4 или 6), маска подсети, протокол, порт, параметры маршрутизации, фрагментация, интерфейс или применяемое определение туннеля. Фильтрация осуществляется на уровне IP, поэтому изменять приложения не требуется.

Туннели позволяют определить конфигурации защиты двух хостов. При этом используются особые параметры защиты, известные только двум связывающимся системам.

На следующем рисунке проиллюстрирован путь пакета от сетевого адаптера до стека IP. После этого вызывается модуль фильтра, определяющий, следует ли принять или отклонить данный пакет. Если указан ИД туннеля, то пакет проверяется на соответствие определению туннеля. Если декапсуляция из формата туннеля выполнена успешно, то пакет передается протоколу следующего уровня. При обработке исходящих пакетов эта процедура выполняется в обратном порядке. При определении отношения пакета к тому или иному туннелю применяются правила фильтрации, однако функция фильтрации может использоваться и в том случае, если пакет передается не через туннель.

На рисунке показана передача сетевых пакетов. Поступающие из сети пакеты принимаются сетевым

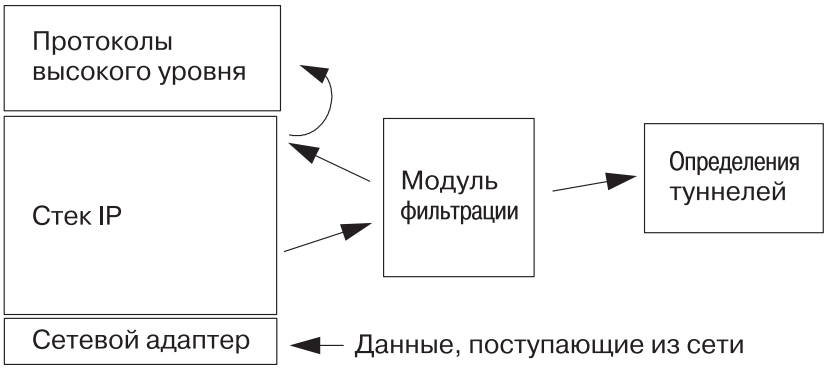


Рисунок 7. Маршрутизация сетевых пакетов

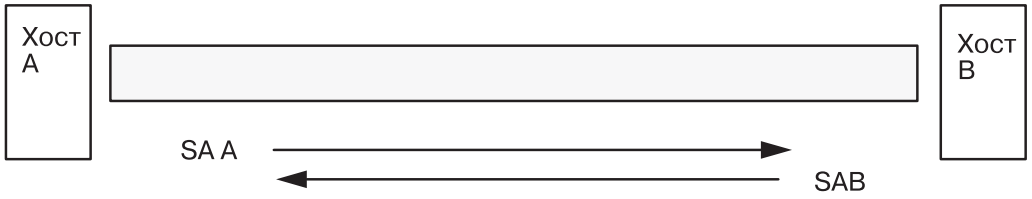
адаптером. Затем пакет передается в стек IP, откуда затем поступает в модуль фильтрации. Из модуля фильтрации пакет передается либо определению туннеля, либо возвращается в стек IP для последующей передачи протоколам более высоких уровней.

Туннели и конфигурации защиты:

Туннели применяются в тех случаях, когда необходимо обеспечить только идентификацию или идентификацию и шифрование данных. Туннели определяются путем создания конфигурации защиты, включающей два хоста. Конфигурация защиты определяет параметры шифрования, алгоритмы идентификации и другие параметры туннелей.

На следующей иллюстрации показан виртуальный туннель между хостами A и B.

На рисунке показан виртуальный туннель между хостами A и B. Конфигурация защиты B обозначена



SA = Конфигурация защиты, состоящая из следующих элементов:

Целевой адрес
Индекс стратегии защиты
Ключ
Алгоритм и формат шифрования
Алгоритм идентификации
Время жизни ключей

Рисунок 8. Установление защищенного туннеля между хостами A и B

стрелкой, направленной от хоста B к хосту A. Конфигурация защиты включает целевой адрес, SPI, ключ, алгоритм шифрования, а также формат, алгоритм идентификации и значение срока действия ключа.

Индекс параметров защиты (SPI) и целевой адрес однозначно идентифицируют уникальную конфигурацию защиты. Эти параметры обязательны для однозначного задания туннеля. Другие параметры, такие как алгоритм шифрования, набор ключей и срок их действия, можно задать явно, либо использовать для них значения по умолчанию.

Особенности туннелей:

Прежде, чем выбирать тип туннеля для защиты IP, следует учесть несколько факторов.

Туннели IKE отличаются от статических туннелей, поскольку процедура настройки стратегии защиты отличается определения конечных точек туннеля.

В IKE применяется двухэтапное согласование. На каждом *этапе* применяются собственные стратегии защиты.

Процесс согласования ключей Internet начинается с установления защищенного канала связи для собственно согласования. Этот этап называется этапом *управления ключами* или *первым этапом*. На этом этапе системы обмениваются идентификационными данными и идентифицируют друг друга с помощью заранее переданных ключей или цифровых сертификатов. На этом этапе создается конфигурация защиты, определяющая способ защищенного обмена информацией, и выбирается способ защиты для второго этапа. В результате создается туннель *IKE* или туннель *первого этапа*.

Второй этап называется этапом *управления данными* или *вторым этапом*, когда туннель IKE применяется для создания конфигураций защиты AH и ESP, обеспечивающих фактическую защиту передаваемых данных. На втором этапе также определяются данные, которые будут применять туннель защиты IP. Например, может быть задана следующая информация:

- Маска подсети
- Диапазон адресов
- Сочетание протокола и номера порта

На этом рисунке показаны два этапа настройки туннеля IKE.

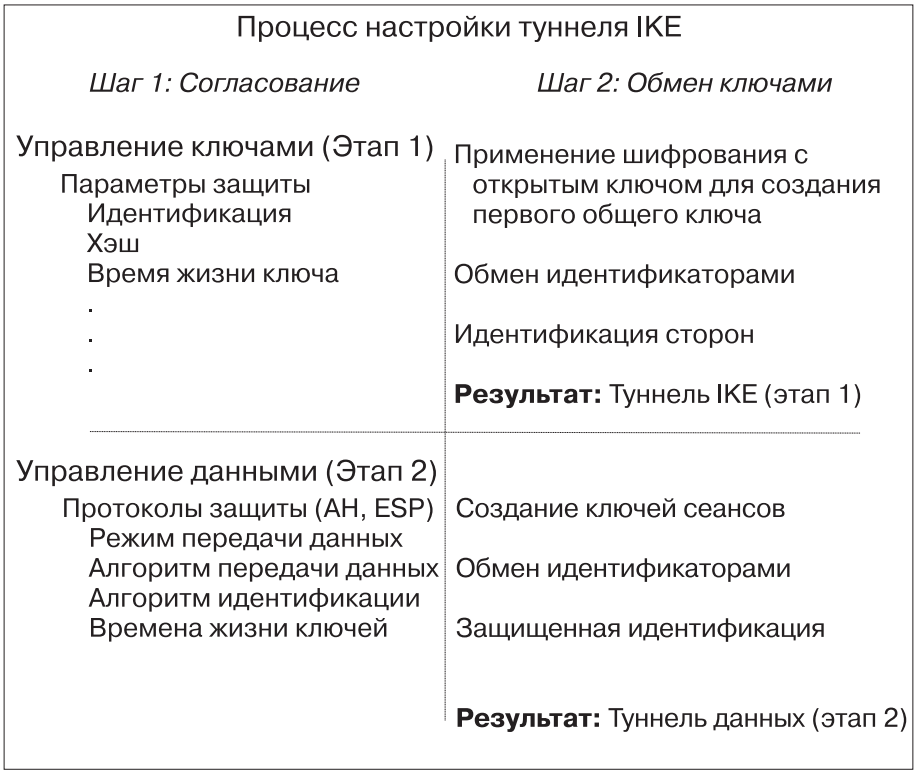


Рисунок 9. Процесс настройки туннеля IKE

Примечание: IKEv2 имеет два этапа. Первый называется *IKE SA* или *phase 1*. Второй — *CHILD SA* или *phase 2*. В отличие от способа создания туннелей в IKEv1, в IKEv2 при создании туннеля "phase 1", туннель "phase 2" активируется автоматически. Настройка туннелей IKEv2 аналогична настройке туннелей IKEv1.

Во многих случаях конечные точки туннеля управления ключами (IKE) совпадают с конечными точками туннеля управления данными (туннеля защиты IP). Конечные точки туннеля IKE соответствуют ИД систем, осуществляющим согласование. Конечные точки туннеля защиты IP описывают тип передаваемых данных, использующих туннель защиты IP. В простых туннелях между хостами, когда все передаваемые между хостами данные защищаются одним и тем же туннелем, конечные точки туннелей первого и второго этапов совпадают. Если согласование выполняется двумя шлюзами, то конечными точками туннеля IKE будут эти два шлюза, а конечными точками туннеля защиты IP будут находящиеся за шлюзами системы или подсети, либо диапазоны адресов пользователей туннеля.

Стратегия и параметры управления ключами:

Вы можете настроить стратегию управления ключами, указав параметры, которые должны применяться при согласовании IKE. Например, существуют стратегии управления ключами для идентификации с помощью режима подписи или заранее подготовленных. На первом этапе пользователь должен определить некоторые свойства защиты управления ключами, которые должны применяться при обмене.

На первом этапе (этапе управления ключами) задаются следующие параметры конфигурации туннеля IKE:

Туннель управления ключами (первый этап)
Имя туннеля IKE. Для каждого туннеля должны быть заданы конечные точки согласования. Это две системы, которые будут передавать и проверять сообщения IKE. Имя туннеля должно описывать его конечные точки, например, `VPN Новосибирск` или `VPN Главный офис`.

Тип идентификатора хоста
Тип идентификатора, который будет применяться при обмене IKE. Для поиска ключей необходимо, чтобы тип и значение идентификатора соответствовали значению, указанному в заранее переданных ключах. Если для поиска значения заранее переданного ключа применяется отдельный идентификатор, то в качестве идентификаторы ключа применяется *ИД хоста*, а в качестве *Типа* - KEY_ID. Тип KEY_ID полезен в том случае, когда на одном хосте хранится несколько заранее полученных значений ключей.

ИД хоста
Значение ИД хоста представляется его IP-адресом, полным доменным именем (FQDN), или пользователем и полным доменным именем (*user@FQDN*). Например: jdoe@studentmail.ut.edu.

IP-адрес
IP-адрес удаленного хоста. Это значение необходимо в том случае, если задан тип ИД KEY_ID, либо если тип ИД хоста невозможно преобразовать в IP-адрес. Например, если имя пользователя невозможно преобразовать с помощью локального сервера имен, то должен быть задан IP-адрес удаленной системы.

Стратегия и параметры управления данными:

Предполагаемые параметры управления данными устанавливаются на первом этапе настройки туннеля IKE. Это те же параметры защиты IP, которые используются в статических туннелях, и определяют тип защиты данных, передаваемых через туннель. Один туннель первого этапа может применяться несколькими туннелями второго этапа.

Следующие типы ИД конечных точек описывают типы данных, применяющих туннель защиты данных IP:

Хост, подсеть или диапазон
Указывает, что передаваемые по туннелю данные предназначены для определенного хоста, подсети или диапазона адресов.

Хост/ИД подсети
Содержит сведения о хосте или подсети локальной и удаленной систем, передающих данные через

туннель. Определяет ИД, передаваемые на втором этапе согласования и правила фильтрации, которые будут применяться в случае успешного согласования.

Маска подсети
Описывает все IP-адреса подсети (например, хост 9.53.250.96 и маска 255.255.255.0).

Начальный IP-адрес диапазона
Задает начальный IP-адрес диапазона адресов, которые будут использовать туннель (например, 9.53.250.96 - это начало диапазона от 9.53.250.96 до 9.53.250.93).

Конечный IP-адрес диапазона
Задает конечный IP-адрес диапазона адресов, которые будут использовать туннель (например, 9.53.250.93 - это конец диапазона от 9.53.250.96 до 9.53.250.93).

Порт Указывает, что данные будут использовать определенный порт (например, 21 или 23).

Протокол
Указывает, что данные будут передаваться с помощью определенного протокола (например, TCP или UDP). Определяет протокол, передаваемый на втором этапе согласования и правила фильтрации, которые будут применяться в случае успешного согласования. Протокол локальной конечной точки должен совпадать с протоколом удаленной конечной точки.

Конечный порт
Описывает конечный порт для передачи данных (например, 100 или 500). По умолчанию конечным портом является 65355.

Ограничение: Для IKEv2 используются только диапазоны адресов IPv4 или IPv6 в качестве селекторов потока данных. Конечный порт применим только для IKEv2 и AIX 6.1 TL 04 и выше.

Выбор типа туннеля:

Выбор статических туннелей или туннелей IKE зависит от того, какие туннели поддерживаются удаленной системой, и какой тип управления ключами более предпочтителен.

По возможности рекомендуется применять туннели IKE, поскольку они обеспечивают стандартное защищенное согласование и обновление ключей. Эти туннели также могут обеспечивать защиту от повторного использования пакетов и применять заголовки IETF ESP и AH. Вы также можете настроить режим подписи для применения цифровых сертификатов.

Если в удаленной системе применяются алгоритмы, требующие использования статических туннелей, то должны применяться статические туннели. Статические туннели гарантируют возможность взаимодействия с большим числом хостов. Поскольку ключи в этом случае статические и их изменение представляет собой не очень простую процедуру, то такой подход обеспечивает менее надежную защиту. Статические туннели могут применяться между хостом, работающим под управлением данной операционной системы, и другой системой, применяющей защиту IP и имеющей общий набор алгоритмов шифрования и идентификации. Многие компании в настоящее время предлагают ключевой MD5 с DES, или HMAC MD5 с DES. Данное подмножество алгоритмов может применяться почти со всеми реализациями защиты IP.

Процедура настройки статических туннелей зависит от того, настраиваете ли первый или второй хост туннеля (параметры второго хоста должны соответствовать параметрам первого). При настройке первого хоста ключи можно генерировать автоматически и использовать алгоритмы по умолчанию. При настройке второго хоста следует по возможности импортировать информацию о туннеле, заданную на первом хосте.

Еще один важный фактор, который следует принимать во внимание, - это то, находится ли удаленная система за брандмауэром. Если это так, то в конфигурацию должна быть включена информация о взаимодействии с брандмауэром.

Применение IKE с DHCP или при динамическом выделении адресов:

В одном из наиболее распространенных сценариев применения защиты IP инициаторами сеансов IKE выступают удаленные системы, которые не имеют постоянного IP-адреса.

Например, такая ситуация возникает в том случае, когда устанавливается соединение с сервером из локальной сети, и для шифрования данных применяется защита IP. В качестве другого примера можно привести случай, когда для идентификации удаленных клиентов, подключающихся к серверу, используется полное имя домена (FQDN) или адрес электронной почты (пользователь@FQDN).

На этапе защиты ключей (Этап 1) подпись RSA является единственным поддерживаемым способом идентификации, если используется основной режим с идентификацией не по IP-адресу. Другими словами, если планируется применять идентификацию с помощью подготовленных ключей, то следует использовать основной или ускоренный режим с IP-адресами в качестве идентификаторов. В действительности при большом числе клиентов DHCP, с которыми необходимо установить туннели IPsec, определение уникальных подготовленных ключей для каждого клиента DHCP становится непрактичным, так что в этом случае рекомендуется применять идентификацию по подписи RSA. В качестве удаленного ИД при определении туннеля можно указать ИД группы, определив таким образом туннель сразу для всех клиентов DHCP (см. пример определения туннеля в файле `/usr/samples/ipsec/group_aix_responder.xml`). ИД группы поддерживается только в реализации IPsec в AIX. При определении группы в нее можно включить любые ИД IKE (в том числе отдельные IP-адреса), FQDN, имя пользователя и FQDN, диапазон или набор IP-адресов и так далее. Созданную группу можно указать в определении туннеля в качестве удаленного узла для первого или второго этапа.

Примечание: При использовании ИД группы туннелю можно назначать только роль отвечающей стороны. Это означает, что данный туннель можно активировать только со стороны клиента DHCP.

Если для этапа защиты данных (второй этап) создается конфигурация защиты для шифрования данных TCP или UDP, то можно настроить общий туннель защиты данных. Таким образом, любой запрос, идентифицированный на первом этапе, будет применять общий туннель защиты данных, если IP-адрес не определен в базе данных. Определению общего туннеля соответствуют любые адреса, поэтому этот туннель используется всегда, когда проверка с помощью общего ключа на первом этапе завершилась успешна.

Определение общего туннеля защиты данных с помощью XML:

Вы можете определить общий туннель управления данными с помощью файла в формате XML, распознаваемого **ikedb**.

Подробная информация об интерфейсе IKE XML и команде **ikedb** приведена в разделе “Настройка туннеля IKE с помощью интерфейса командной строки” на стр. 235. Общие туннели управления данными применяются с DHCP. В формате XML применяется тег IPSecTunnel. Этот туннель также называется *туннелем второго этапа*. *Общий туннель управления данными* не является туннелем в полном смысле этого слова, а представляет собой конфигурацию защиты IPSecProtection, применяемую в том случае, если полученное через определенный туннель управления ключами сообщение управления данными не соответствует ни одному из туннелей управления данными для этого туннеля управления ключами. Он применяется только в том случае, если система AIX работает в режиме ответов. Указывать IPSecProtection для общего туннеля управления данными необязательно.

Общий туннель управления данными определяется в элементе IKEProtection. Для этого применяются атрибуты XML *IKE_IPSecDefaultProtectionRef* и *IKE_IPSecDefaultAllowedTypes*.

Во-первых, необходимо определить конфигурацию защиты IPSecProtection, которая будет применяться по умолчанию при отсутствии других туннелей управления ключами IPSecTunnels. IPSecProtection по умолчанию должна иметь атрибут IPSec_ProtectionName, начинающийся с символов `_defIPSprot_`.

Теперь перейдем к IKEProtection, которая будет применяться с IPSecProtection по умолчанию. Укажите атрибут **IKE_IPSecDefaultProtectionRef**, содержащий имя IPSec_Protection по умолчанию.

Также следует указать в IKEProtection значение атрибута **IKE_IPSecDefaultAllowedTypes**. Это может быть одно или несколько значений из следующего списка (значения должны разделяться пробелами):

```
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
```

Эти значения соответствуют указанным инициатором типам ИД. При согласовании IKE фактические ИД игнорируются. Указанная конфигурация защиты IPSecProtection применяется в том случае, если атрибут **IKE_IPSecDefaultAllowedTypes** содержит строку, начинающуюся с символов `Local_`, и соответствующую типу локального ИД инициатора, и строку, начинающуюся с символов `Remote_`, и соответствующую удаленного ИД инициатора. Другими словами, для применения `IPSec_Protection` в каждом атрибуте `IKE_IPSecDefaultAllowedTypes` должно быть определено по крайней мере одно значение **Local_** и по крайней мере одно значение Remote_.

Пример общего туннеля защиты данных:

Туннель защиты данных можно использовать для отправления сообщений системе.

На втором этапе (этап управления данными) инициатор отправляет системе AIX следующее сообщение:

```
тип локального ИД:    IPV4_Address
локальный ИД:         192.168.100.104

тип удаленного ИД:    IPV4_Subnet
удаленный ИД:         10.10.10.2
удаленная маска сети: 255.255.255.192
```

В системе AIX нет туннеля защиты данных, соответствующего этим ИД. Однако в ней есть IPSecProtection со следующими атрибутами:

```
IKE_IPSecDefaultProtectionRef="_defIPSprot_protection4"
IKE_IPSecDefaultAllowedTypes="Local_IPV4_Address
                              Remote_IPV4_Address
                              Remote_IPV4_Subnet
                              Remote_IPV4_Address_Range"
```

Тип локального ИД в поступающем сообщении (IPV4_Address) соответствует одному из допустимых значений Local_ (Local_IPV4_Address). Кроме того, удаленный ИД в сообщении (IPV4_Subnet) соответствует значению Remote_IPV4_Subnet. Таким образом, согласование туннеля управления данными будет продолжено с использованием `_defIPSprot_protection4` в качестве IPSecProtection.

Файл `/usr/samples/ipsec/default_p2_policy.xml` представляет собой пример файла XML с определением IPSecProtection.

Настройка туннелей IKE (Обмен Internet-ключами)

Можно настроить туннели IKE с помощью программы SMIT и командной строки.

Настройка туннеля IKE с помощью интерфейса SMIT:

Для настройки туннелей IKE и выполнения основных операций над базой данных может применяться интерфейс SMIT.

SMIT применяет для добавления, удаления и изменения определений туннелей IKE базовые команды XML. SMIT применяется для быстрой настройки туннелей IKE и позволяет ознакомиться с примерами синтаксиса XML, применяемого для создания определений туннелей IKE. Меню IKE SMIT также позволяют сохранять, восстанавливать и инициализировать базу данных IKE.

Для настройки туннеля IPv4 введите команду быстрого доступа **smitty ike4**. Для настройки туннеля IPv6 введите команду быстрого доступа **smitty ike6**. Операции над базой данных IKE описаны в меню Дополнительная настройка защиты IP-пакетов.

Настройка туннеля IKE с помощью интерфейса командной строки:

Команда **ikedb** позволяет пользователю получить, обновить, удалить, импортировать и экспортировать информацию из базы данных IKE с помощью интерфейса XML.

Команда **ikedb** позволяет пользователю записать (поместить) или считать (получить) данные из базы данных IKE. Для ввода и вывода применяется формат XML. Формат файла XML задается связанным с ним Определением типа документа (DTD). Команда **ikedb** позволяет пользователю просмотреть DTD, применяемый для проверки файла XML при записи данных. Единственное изменение, которое можно внести в DTD - добавление деклараций экземпляра с помощью флага **-e**. Любая декларация с внешним DOCTYPE во входном файле XML будет проигнорирована, а декларация с внутренним DOCTYPE может вызвать ошибку. Правила анализа файла XML с помощью DTD описаны в стандарте XML. Файл `/usr/samples/ipsec` содержит пример типичного файла XML, определяющего стандартные туннели. Сведения о синтаксисе приведены в описании команды **ikedb** в разделе *Справочник по командам*.

Команда **ike** позволяет запускать, останавливать и отслеживать состояние туннелей IKE. Команда **ike** может также применяться для активации, удаления туннеля и просмотра списка туннелей. Сведения о синтаксисе приведены в описании команды **ike** в разделе *Справочник по командам*.

Применение команд **ike**, **ikedb** и некоторых других для настройки и проверки состояния туннеля IKE продемонстрировано в следующих примерах:

1. Для запуска процедуры согласования туннеля (его *активации*) или разрешения согласования в режиме отвечающей стороны (в зависимости от указанной роли), введите команду **ike** с номером туннеля:

   ```
   # ike cmd=activate numlist=1
   ```

   Также можно указать удаленный идентификатор или IP-адреса:

   ```
   # ike cmd=activate remid=9.3.97.256
   # ike cmd=activate ipaddr=9.3.97.100, 9.3.97.256
   ```

   Поскольку согласование может занять несколько секунд, команды завершаются сразу после его начала.
2. Для просмотра состояния туннеля введите следующую команду **ike**:

   ```
   # ike cmd=list
   ```

   Ниже приведен пример вывода этой команды:

   ```
   Этап 1  ИД туннеля      [1]
   Этап 2  ИД туннеля      [1]
   ```

   В выводе показаны туннели этапов 1 и 2, активные в момент запуска команды.
3. Для получения подробной информации о туннеле введите следующую команду **ike**:

   ```
   # ike cmd=list verbose
   ```

   Ниже приведен пример вывода этой команды:

```
Этап 1  ИД Туннеля      1
Тип локального ИД:      Fully_Qualified_Domain_Name
Локальный ИД:           bee.austin.ibm.com
Тип удаленного ИД:      Fully_Qualified_Domain_Name
Удаленный ИД:           ipsec.austin.ibm.com
Режим:                  Ускоренный
Стратегия защиты:       BOTH_AGGR_3DES_MD5
Роль:                   Инициатор
Алгоритм шифрования:    3DES-CBC
Алгоритм идентификации: Подготовленный ключ
Алгоритм хэширования:   MD5
Срок действия ключа:    28800 секунд
Ресурс ключа:           0 Кб
Остаток срока действия: 28737 секунд
Остаток ресурса:        0 Кб
Перекр. обновл. ключа:  5%
Срок действия туннеля:  2592000 секунд
Ресурс туннеля:         0 Кб
Остаток срока действия: 2591937 секунд
Состояние:              Активен

Этап 2  ИД Туннеля      1
Тип локального ИД:      IPv4_Address
Локальный ИД:           10.10.10.1
Маска локальной подсети:нд
Локальный порт:         любой
Локальный протокол:     все
Тип удаленного ИД:      IPv4_Address
Удаленный ИД:           10.10.10.4
Маска удаленной подсети:нд
Удаленный порт:         любой
Удаленный протокол:     все
Режим:                  Oakley_quick
Стратегия защиты:       ESP_3DES_MD5_SHA_TUNNEL_NO_PFS
Роль:                   Инициатор
Алгоритм шифрования:    ESP_3DES
Преобразование AH:      нд
Алгоритм идентификации: HMAC-MD5
PFS:                    Нет
Срок действия SA:       600 секунд
Ресурс SA:              0 Кб
Остаток срока действия: 562 секунд
Остаток ресурса:        0 Кб
Перекр. обновл. ключа:  15%
Срок действия туннеля:  2592000 секунд
Ресурс туннеля:         0 Кб
Остаток срока действия: 2591962 секунд
Связано туннелей P1:    0
Режим инкапсуляции:     ESP_tunnel
Состояние:              Активен
```

4. Для просмотра правил фильтрации из таблицы динамической фильтрации для нового туннеля IKE введите следующую команду **lsfilt**:

   ```
   # lsfilt -d
   ```

   Ниже приведен пример вывода этой команды:

   ```
   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 *** Расположение правила динамической фильтрации *** 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

   *** Динамическая таблица ***

   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
```

Этот пример соответствует системе с одним туннелем IKE и отсутствием других туннелей. Расположение правила динамической фильтрации (строка номер 2 в данном примере вывода статической таблицы) задает расположение правила относительно других пользовательских правил и может изменяться пользователем. Правила создаются и вставляются в таблицу автоматически при активации туннелей. Возможен просмотр, но не изменение таких правил.

5. Для включения регистрации укажите в опции ведения журнала для правила #2 значение yes, и введите следующую команду **chfilt**:

   ```
   # chfilt -v 4 -n 2 -l y
   ```

   Дополнительная информация о регистрации потока данных IKE приведена в разделе “Средства ведения протокола” на стр. 260.

6. Для деактивации туннеля введите следующую команду **ike**:

   ```
   # ike cmd=remove numlist=1
   ```

7. Для просмотра определений туннеля введите команду **ikedb**:

   ```
   # ikedb -g
   ```

8. Для помещения в базу данных IKE определений из файла XML, созданного в удаленной системе, с перезаписью существующих объектов, имеющих в базе данных то же имя, введите следующую команду **ikedb**:

   ```
   # ikedb -pFs peer_tunnel_conf.xml
   ```

   `peer_tunnel_conf.xml` - это имя файла XML, созданного в удаленной системе.

9. Для получения определения туннеля этапа 1 с именем *tunnel_sys1_and_sys2* и всех связанных с ним туннелей этапа 2 введите следующую команду **ikedb**:

   ```
   # ikedb -gr -t IKETunnel -n tunnel_sys1_and_sys2
   ```

10. Для удаления из базы данных всех подготовленных ключей введите следующую команду **ikedb**:

    ```
    # ikedb -d -t IKEPresharedKey
    ```

Общие сведения о поддержке групп туннелей IKE приведены в разделе “Поддержка групп”. Команда **ikedb** позволяет определять группы из командной строки.

AIX IKE и среда Linux:

В AIX есть возможность настройки туннеля IKE с помощью файлов конфигурации Linux.

Для настройки туннеля IKE AIX с помощью файлов настройки Linux используйте команду **ikedb** с флагом **-c** (опция преобразования), которая позволяет использовать файлы настройки Linux `/etc/ipsec.conf` и `/etc/ipsec.secrets` как определения туннеля IKE. Команда **ikedb** анализирует файлы конфигурации Linux, создает файл XML и (необязательно) добавляет определения туннеля из файла XML в базу данных IKE. После этого для просмотра определений туннелей можно воспользоваться командой **ikedb -g**.

Поддержка групп:

Защита IP-пакетов поддерживает объединение идентификаторов IKE в определении туннеля для связи нескольких идентификаторов с одной стратегией защиты вместо создания нескольких независимых определений туннеля.

Объединение в группы особенно полезно при настройке соединений с несколькими удаленными хостами, поскольку позволяет избежать настройки и управления несколькими определениями туннелей. Кроме того, все изменения достаточно будет вносить однократно в одну стратегию защиты.

Перед использованием в определении туннеля группа должна быть определена. Размер группы ограничен 1 Кб. Со стороны инициатора группы могут применяться в качестве удаленных идентификаторов только в определениях туннелей защиты данных. Отвечающая сторона может применять группы в качестве удаленных идентификаторов как в определениях туннелей защиты данных, так и в определениях туннелей защиты ключей.

Группа состоит из имени группы, списка типов идентификаторов и самих идентификаторов IKE. Все идентификаторы могут быть одного типа, либо относиться к одному из следующих типов:

- адрес IPv4
- адрес IPv6
- домен
- пользователь@домен
- типы DN X500

При согласовании конфигурации защиты идентификаторы в группе просматриваются последовательно до первого соответствия.

Информация об определении групп из командной строки приведена в разделе “Настройка туннеля IKE с помощью интерфейса командной строки” на стр. 235.

Сценарии настройки туннеля IKE:

Следующие сценарии описывают стандартные ситуации, с которыми встречаются большинство пользователей при настройке туннелей. Эти сценарии соответствуют связи с филиалом, деловым партнером и удаленным сотрудником.

- В случае филиала существуют две физически удаленные друг от друга защищенные сети, которые необходимо связать друг с другом. В этом примере шлюзы связаны туннелем, по которому передаются все данные для другой сети. На противоположной стороне туннеля пакеты данных расформируются и передаются по внутренней сети в открытом виде.

  На первом этапе согласования IKE создается общая для двух шлюзов конфигурация защиты IKE. Поток данных, передаваемый по туннелю, - это поток данных между двумя подсетями, поэтому на втором этапе согласования применяются идентификаторы подсетей. После указания для туннеля стратегии защиты и параметров туннеля создается номер туннеля. Затем туннель можно активировать командой **ike**.
- В случае делового партнера сети не считаются доверенными, поэтому рекомендуется ограничить удаленный доступ небольшим числом хостов за брандмауэром. В этом случае создается туннель для защищенной передачи данных между двумя конкретными хостами. В качестве протокола туннеля второго этапа применяется AH или ESP. Этот туннель хост-хост проходит через туннель шлюз-шлюз.
- В случае доступа удаленного сотрудника туннели создаются по требованию для обеспечения максимальной защиты. IP-адреса могут меняться, поэтому для идентификации могут применяться полные имена доменов, возможно, с указанием пользователя (*user@*). В качестве альтернативы для связи ключа с идентификатором хоста может применяться KEYID.

Цифровые сертификаты и Диспетчер ключей

Цифровые сертификаты - это идентификаторы общих ключей, позволяющие проверить личность отправителя или получателя зашифрованного сообщения.

Средства защиты протокола IP поддерживают цифровые сертификаты и *шифрование с общим ключом* (*асимметричное шифрование*), идея которого заключается в том, что шифрование данных осуществляется с

помощью общего ключа, известного всем, а дешифрование - с помощью личного ключа получателя. Общий и личный ключ образуют уникальную пару. *Пары ключей* - это большие строки данных, выступающие в роли ключей при шифровании.

В шифровании с открытым ключом получатель данных передает свой общий ключ всем, от кого он намерен получать зашифрованные сообщения. Ко всем отправляемым сообщениям добавляются цифровые подписи, создаваемые с помощью личных ключей отправителей. Получатели проверяют подлинность подписей с помощью соответствующих общих ключей. Сначала сообщение дешифруется личным ключом получателя, а затем получатель проверяет подпись отправителя с помощью общего ключа отправителя.

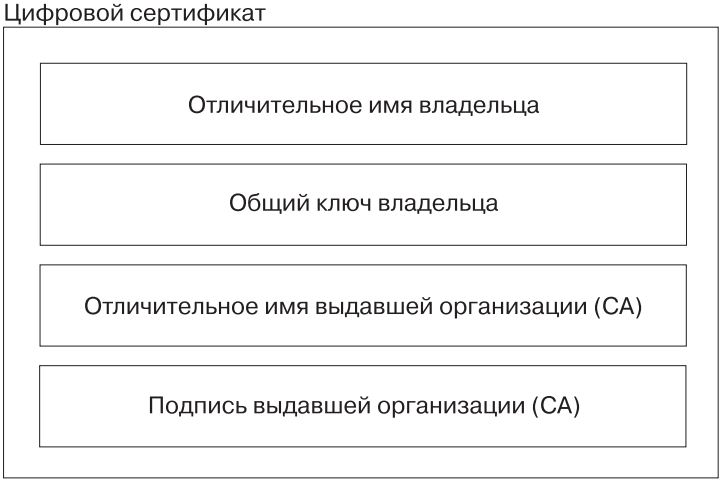
В системе шифрования участвуют независимые организации или лица, называющиеся *сертификатными компаниями* (*CA*), которые выдают цифровые сертификаты после проверки личности заявителя. Получатели сообщений по своему усмотрению выбирают сертификатные компании. Срок действия сертификатов ограничен, по окончании этого срока сертификат должен быть заменен.

AIX предоставляет инструмент Администратор ключей, который управляет цифровыми сертификатами. В следующих разделах приведены общие сведения о цифровых сертификатах.

Формат цифровых сертификатов:

В цифровых сертификатах содержатся данные об их владельцах и о сертификатных компаниях, выдавших эти сертификаты. Структура цифрового сертификата проиллюстрирована на следующем рисунке.

На этом рисунке показаны четыре компонента цифрового сертификата. Это отличительное имя владельца,



Состав цифрового сертификата

Рисунок 10. Содержимое цифрового сертификата

общий ключ владельца, отличительное имя сертификатной компании и подпись сертификатной компании.

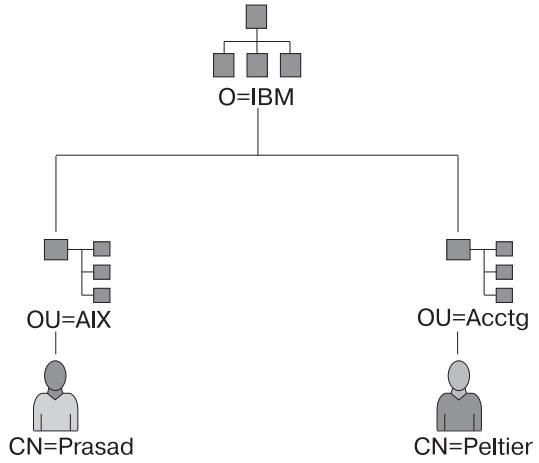
Подробное описание этих компонентов приведено ниже:

Отличительное имя владельца

Имя владельца сертификата и его расположение в иерархическом каталоге (дереве). На следующем рисунке приведен пример простого дерева: сертификат принадлежит сотруднику Петрову в контексте country=RU (страна - Россия), organization=IJK (фирма - IJK), lower organization=SERV (отдел - SERV). Полное отличительное имя выглядит следующим образом:

```
/C=RU/O=IJK/OU=SERV/CN=petrov.serv.ijk.ru
```

На этом рисунке показан иерархический каталог, начинающийся с позиции O=IJK и состоящий из



Пример получения отличительного имени из дерева каталогов

Рисунок 11. Пример получения отличительного имени из дерева

двух подкаталогов. Один из подкаталогов называется OU=AIX, а второй - OU=Acctg. В каждом подкаталоге находится один объект. В одном подкаталоге это CN=Petrov, а во втором - CN=Nikolaev.

Общий ключ владельца сертификата
Используется для дешифрования цифровых подписей владельца сертификата.

Альтернативное имя
Любой альтернативный идентификатор: IP-адрес, почтовый адрес, полное доменное имя и т.п.

Дата выдачи
Дата выдачи цифрового сертификата.

Срок действия
Дата окончания срока действия цифрового сертификата.

Отличительное имя сертификатной компании
Отличительное имя сертификатной компании.

Цифровая подпись сертификатной компании
Подпись, позволяющая проверить подлинность сертификатной компании.

Особенности работы цифровых сертификатов:

Одного цифрового сертификата недостаточно для надежной идентификации.

Цифровой сертификат позволяет только идентифицировать своего владельца с помощью общего ключа, применяемого для дешифрования цифровых подписей владельца. Общий ключ можно распространять без ограничений, потому что для дешифрования данных, зашифрованных с помощью общего ключа, нужен личный ключ. Личный ключ нужно хранить в секрете. Если кому-либо станет известен личный ключ сертификата, он сможет расшифровывать все сообщения, зашифрованные с помощью соответствующего общего ключа. Без личного ключа злоупотребить сертификатом невозможно.

Сертификатные компании и списки надежности:

Цифровому сертификату можно доверять ровно настолько, насколько вы доверяете сертификатной компании, которая его выдала.

Степень доверия к сертификатной компании должна зависеть от правил, которыми эта компания руководствуется при выдаче сертификатов. Решение вопроса о доверии и недоверии к сертификатным компаниям - личное дело каждого пользователя или организации.

С помощью диспетчера ключей можно создавать и подписывать собственные сертификаты, что удобно для организации защищенного обмена данными в рамках небольших компаний.

В этом случае нужно передать общий ключ сертификатной компании, созданной с помощью диспетчера ключей, всем сотрудникам. Сам по себе факт получения цифрового сертификата не гарантирует того, что его владелец - действительно тот, за кого он себя выдает. Для проверки личности отправителя нужен общий ключ сертификатной компании, выдавшей сертификат. Если у вас нет копии общего ключа сертификатной компании, его можно получить из соответствующего цифрового сертификата.

Списки аннулированных сертификатов:

Цифровые сертификаты рассчитаны на то, что они будут использоваться в течение всего срока действия. Однако они могут быть аннулированы и ранее.

Такая необходимость может возникнуть в случае, если сотрудник покидает компанию до истечения срока действия сертификата, или если личный ключ сертификата становится известен посторонним лицам. Для того чтобы аннулировать сертификат, нужно сообщить сертификатной компании о причинах такого решения. Аннулирование сертификата заключается в том, что сертификатная компания добавляет его номер в список аннулированных сертификатов.

Списки аннулированных сертификатов подписываются сертификатными компаниями и регулярно распространяются. Списки аннулированных сертификатов можно загружать с серверов сертификатных компаний по протоколам HTTP и LDAP. В каждом списке указана дата его создания и дата следующего обновления. Для каждого аннулированного сертификата указан его номер.

Если вы будете применять цифровые сертификаты как средство идентификации для работы туннеля IKE, то можете настроить автоматическую проверку их действительности с помощью опции Подпись RSA с проверкой CRL. Если включена опция Проверка CRL, то список аннулированных сертификатов будет загружаться и просматриваться в каждом сеансе создания туннеля с защитой ключей.

> **Примечание:** Для применения этой опции необходимо, чтобы системе был доступен сервер SOCKS (версия 4 для серверов HTTP) и (или) сервер LDAP. Если известно, какой сервер SOCKS или LDAP используется для получения CRL, то можно добавить его в файл `/etc/isakmpd.conf`

Применение цифровых сертификатов в приложениях Internet:

Приложения Internet, в которых используются схемы шифрования с открытым ключом, применяют цифровые сертификаты в качестве источников общих ключей.

В настоящее время шифрование с открытым ключом применяется во многих приложениях:

Виртуальные частные сети (VPN)

Виртуальные частные сети (*защищенные туннели*) - это защищенные каналы связи между брандмауэрами, применяемые для организации защищенного соединения между двумя защищенными сетями по открытой (незащищенной) сети. Все данные, передаваемые между защищенными сетями, проходят по открытой сети в зашифрованном виде.

Для организации туннеля применяются протоколы, соответствующие стандартам IKE и защиты IP. Эти протоколы предназначены для создания защищенных зашифрованных соединений между удаленными хостами (например, домашними компьютерами сотрудников организации) и защищенной сетью или защищенным хостом.

Протокол SSL

Протокол SSL применяется для обеспечения защиты и целостности передаваемых данных. Этот

протокол применяется Web-серверами - в защищенных соединениях между Web-серверами и Web-браузерами, протоколом LDAP - в в защищенных соединениях между клиентами LDAP и серверами LDAP, продуктом Host-on-Demand V.2 - в соединениях между клиентами и главной системой. Протокол SSL использует цифровые сертификаты для обмена ключами, идентификации серверов и, при необходимости, идентификации клиентов.

Защищенная электронная почта
В ряде стандартов электронной почты (например, PEM или S/MIME) предусмотрена возможность применения цифровых сертификатов для создания цифровых подписей и обмена ключами с целью шифрования почтовых сообщений.

Цифровые сертификаты и запросы на сертификаты:

Для того чтобы получить цифровой сертификат от сертификатной компании, ей нужно отправить *запрос на сертификат*.

Подписанный цифровой сертификат содержит отличительное имя и общий ключ своего владельца, а также отличительное имя и подпись сертификатной компании. Если цифровой сертификат подписан его владельцем, то в нем содержатся отличительное имя владельца, его общий ключ и подпись.

В запросе нужно указать отличительное имя будущего владельца, его общий ключ и подпись. Сертификатная компания проверит цифровую подпись с помощью общего ключа, указанного в сертификате, с целью убедиться в том, что:

- Запрос не был подделан во время передачи от запрашивающего лица в сертификатную компанию.
- У запрашивающего лица есть личный ключ, соответствующий общему ключу, указанному в запросе на сертификат.

Кроме того, сертификатная компания обязана каким-либо способом проверить личность запрашивающего сертификат. Правила проверки устанавливаются самими компаниями и сильно варьируются по своей жесткости.

Инструмент Диспетчер ключей:

Диспетчер ключей - это программа, предназначенная для работы с цифровыми сертификатами. Она входит в набор файлов `gskkm.rte`, поставляемом в пакете расширения.

Для настройки системы обработки цифровых сертификатов и подписей нужно, как минимум, выполнить задачи 1, 2, 3, 4, 6 и 7. После этого необходимо создать туннель IKE и установить для него правила, предполагающие идентификацию на основе цифровых подписей RSA.

Базу данных ключей можно создать и настроить, вызвав команду certmgr для открытия инструмента Администратор ключей из командной строки.

В этом разделе приведены инструкции по выполнению следующих операций с помощью Диспетчера ключей:

Создание базы ключей:

База ключей нужна для применения цифровых сертификатов в VPN. Средства защиты IP в AIX работают с базами данных в формате *.kdb.

Диспетчер ключей поддерживает следующие типы сертификатов CA:

- RSA Secure Server Certification Authority
- Thawte Personal Premium Certification Authority
- Thawte Personal Freemail Certification Authority
- Thawte Personal Basic Certification Authority

- Thawte Personal Server Certification Authority
- Thawte Server Certification Authority
- Verisign Class 1 Public Primary Certification Authority
- Verisign Class 2 Public Primary Certification Authority
- Verisign Class 3 Public Primary Certification Authority
- Verisign Class 4 Public Primary Certification Authority

По умолчанию клиенты могут подключаться к серверам с сертификатами, выданными любой из этих компаний. После создания базы ключей можно сразу подключаться к таким серверам.

Если вы хотите применять сертификаты компании, которой нет в этом списке, нужно запросить сертификат подписи этой компании и добавить его в базу ключей. Дополнительная информация приведена в разделе “Добавление базового сертификата CA”.

Для создания базы ключей с помощью команды **certmgr** выполните следующие действия:

1. Запустите Диспетчер ключей, введя команду:
   ```
   # certmgr
   ```
2. В окне База ключей откройте меню Файл и выберите пункт **Создать**.
3. Оставьте значение по умолчанию (Файл базы ключей CMS) в поле **Тип базы ключей**.
4. Укажите следующее значение в поле **Имя файла**:
   ```
   ikekey.kdb
   ```
5. Укажите следующее значение в поле **Расположение**:
   ```
   /etc/security
   ```

   Примечание: База ключей должна храниться в файле `ikekey.kbd` в каталоге `/etc/security`. Другие файлы не поддерживаются средствами защиты IP.
6. Нажмите кнопку **OK**. Появится окно **Пароль**.
7. Введите пароль в поле **Пароль** и в поле **Подтверждение пароля**.
8. Для изменения числа дней до истечения срока действия введите нужное значение в днях в поле **Задать срок действия?**. По умолчанию срок действия пароля составляет 60 дней. Для задания неограниченного срока действия пароля выключите поле **Задать срок действия?**.
9. Для сохранения зашифрованной версии пароля в бумажнике включите поле **Сохранить пароль в файле?** и выберите Да.

   Примечание: Для применения цифровых сертификатов со средствами защиты IP сохранение пароля в файле обязательно.
10. Нажмите кнопку **OK**. Появится окно подтверждения.
11. Снова нажмите **OK** для возврата в окно Работа с ключами IBM. Теперь можно перейти к выполнению других задач или закончить работу с этой программой.

Добавление базового сертификата CA:

Когда вы получите базовый сертификат от сертификатной компании, его нужно добавить в базу данных.

Большинство базовых сертификатов поставляется в файлах с расширением *.arm, например:
```
cert.arm
```

Для того чтобы добавить базовый сертификат CA в базу данных, выполните следующие действия:

1. Запустите Диспетчер ключей, если он еще не запущен:
   ```
   # certmgr
   ```
2. В главном окне откройте меню Файл базы ключей и выберите пункт **Открыть**.

3. Выберите файл базы ключей, в который нужно добавить базовый сертификат CA, и нажмите кнопку **Открыть**.
4. Введите пароль и нажмите кнопку **OK**. Если пароль указан правильно, то вновь отобразится окно Работа с ключами IBM. В заголовке окна будет показано имя выбранного файла базы ключей. Это означает, что файл открыт и готов к работе.
5. Войдите в меню **Сертификаты** и выберите пункт **Сертификаты CA** from the list.
6. Нажмите кнопку **Добавить**.
7. Выберите нужный тип данных в выпадающем списке **Тип данных**, например:

   ```
   Данные ASCII в формате Base64
   ```
8. Введите имя и расположение файла базового сертификата или нажмите кнопку **Обзор** и выберите файл из списка.
9. Нажмите кнопку **OK**.
10. Введите метку для базового сертификата CA, например, `Пробный базовый сертификат CA`, затем нажмите **OK**. Вновь появится окно **Работа с ключами**. В поле **Сертификаты CA** будет показана метка вновь добавленного базового сертификата CA. Теперь можно перейти к выполнению других задач или закончить работу с этой программой.

Настройка параметров надежности:

По умолчанию все установленные сертификаты CA считаются надежными. При необходимости параметры надежности можно изменить.

Для изменения этого параметра выполните следующие действия:

1. Запустите Диспетчер ключей, если он еще не запущен:

   ```
   # certmgr
   ```
2. В главном окне откройте меню **Файл базы ключей** и выберите **Открыть**.
3. Выберите файл базы ключей, в котором нужно изменить стандартный сертификат, и нажмите кнопку **Открыть**.
4. Введите пароль и нажмите кнопку **OK**. Если пароль указан правильно, то вновь отобразится окно **Работа с ключами IBM**. В заголовке окна будет показано имя выбранного файла базы ключей. Это означает, что файл открыт.
5. Войдите в меню **Сертификаты** и выберите пункт **Сертификаты CA** from the list.
6. Выберите нужный сертификат и дважды щелкните на нем или нажмите кнопку **Показать (изменить)**. Появится окно **Информация о ключе**.
7. Если вы хотите сделать данный сертификат надежным базовым сертификатом, включите переключатель **Надежный базовый сертификат** и нажмите кнопку **OK**. Если вы не считаете данный сертификат надежным, снимите отметку с этого поля и нажмите кнопку **OK**.
8. Нажмите **OK** в окне **Сертификаты CA**. Вновь отобразится окно **Работа с ключами IBM**. Теперь можно перейти к выполнению других задач или закончить работу с этой программой.

Удаление базового сертификата CA:

Если вы хотите прекратить работу с сертификатами, выданными определенной сертификатной компанией, нужно удалить базовый сертификат этой компании из базы ключей.

Примечание: Перед удалением базового сертификата рекомендуется сохранить его резервную копию на случай, если в дальнейшем потребуется возобновить работу с этой компанией.

Для того чтобы удалить базовый сертификат CA из базы данных, выполните следующие действия:

1. Запустите Диспетчер ключей, если он еще не запущен:

   ```
   # certmgr
   ```

2. В главном окне откройте меню **Файл базы ключей** и выберите пункт **Открыть**.
3. Выберите файл базы ключей, из которого нужно удалить базовый сертификат CA, и нажмите кнопку **Открыть**.
4. Введите пароль и нажмите кнопку **ОК**. Если пароль указан правильно, вновь появится окно **Работа с ключами**. В заголовке окна будет показано имя выбранного файла базы ключей. Это означает, что файл открыт и готов к работе.
5. Войдите в меню **Сертификаты** и выберите пункт **Сертификаты CA** from the list.
6. Выберите нужный сертификат и нажмите кнопку **Удалить**. Появится окно **Подтверждение**.
7. Нажмите кнопку **Да**. Вновь отобразится окно **Работа с ключами IBM**. Удаленный базовый сертификат больше не будет отображаться в поле **Сертификаты CA**. Теперь можно перейти к выполнению других задач или закончить работу с этой программой.

Запрос сертификата:

Для того чтобы получить цифровой сертификат, нужно создать запрос с помощью Диспетчера ключей и отправить его в сертификатную компанию. Запросы сохраняются в файлах в формате PKCS#10. После этого сертификатная компания проверяет вашу личность и отправляет вам цифровой сертификат.

Для того чтобы запросить цифровой сертификат, выполните следующие действия:

1. Запустите Диспетчер ключей, если он еще не запущен:
   ```
   # certmgr
   ```
2. В главном окне откройте меню **Файл базы ключей** и выберите пункт **Открыть**.
3. Выберите файл базы ключей `/etc/security/ikekey.kdb`, из которого будет формироваться запрос, и нажмите **Открыть**.
4. Введите пароль и нажмите кнопку **ОК**. Если пароль указан правильно, то вновь отобразится окно **Работа с ключами IBM**. В заголовке окна будет показано имя выбранного файла базы ключей. Это означает, что файл открыт и готов к работе.
5. Выберите пункт меню **Создать > Новый запрос на сертификат**.
6. Нажмите кнопку **Создать**.
7. Укажите метку ключа для своего сертификата, например:
   ```
   keytest
   ```
8. Укажите общее имя (по умолчанию это имя хоста) и организацию, затем выберите страну. В остальных полях можно оставить значения по умолчанию или указать другие значения.
9. Укажите альтернативное имя. В разделе Альтернативное имя можно указать почтовый адрес, IP-адрес или доменное имя в системе DNS. Для туннельных IP-адресов нужно обязательно указать IP-адрес туннеля. Для туннелей *user@FQDN* нужно указать почтовый адрес. Для туннелей FQDN в поле Имя в DNS нужно указывать полное доменное имя (например, *хост.компания*.com).
10. Укажите имя файла в нижней части окна, например:
    ```
    certreq.arm
    ```
11. Нажмите кнопку **ОК**. Появится окно подтверждения с сообщением о том, что создан запрос на выдачу нового сертификата.
12. Нажмите кнопку **ОК**. Вновь отобразится окно **Работа с ключами IBM**. В поле **Запросы на личные сертификаты** будет показана метка ключа нового запроса.
13. Отправьте файл с запросом в сертификатную компанию. Теперь можно перейти к выполнению других задач или закончить работу с этой программой.

Получение и установка сертификата:

Когда вы получите сертификат от сертификатной компании, его нужно будет установить в базе ключей, из которой был создан запрос.

Для того чтобы установить сертификат, выполните следующие действия:

1. Запустите Диспетчер ключей, если он еще не запущен:

   ```
   # certmgr
   ```

2. В главном окне откройте меню **Файл базы ключей** и выберите пункт **Открыть**.
3. Выберите файл базы данных, который применялся для создания запроса на сертификат, и нажмите кнопку **Открыть**.
4. Введите пароль и нажмите кнопку **ОК**. Если пароль указан правильно, то вновь отобразится окно Работа с ключами IBM. В заголовке окна будет показано имя выбранного файла базы ключей. Это означает, что файл открыт и готов к работе.
5. Войдите в меню **Сертификаты** и выберите **Запросы на личные сертификаты**.
6. Нажмите кнопку **Получить**, чтобы вновь полученный сертификат добавился в базу данных.
7. В выпадающем списке **Тип данных** выберите тип данных сертификата. По умолчанию применяется значение **Данные ASCII в формате Base64**.
8. Введите имя и расположение файла базового сертификата или нажмите кнопку **Обзор** и выберите файл из списка.
9. Нажмите кнопку **ОК**.
10. Укажите метку для нового сертификата, например:

    ```
    VPN Branch Certificate
    ```

11. Нажмите кнопку **ОК**. Вновь отобразится окно **Работа с ключами IBM**. В поле **Личные сертификаты** будет показан новый сертификат. Теперь можно перейти к выполнению других задач или закончить работу с этой программой. Если при загрузке сертификата возникнет ошибка, то убедитесь в том, что содержимое файла начинается строкой `---BEGIN CERTIFICATE---` и заканчивается строкой `---END CERTIFICATE---`.

    Например:

    ```
    -----BEGIN CERTIFICATE-----
    ajdkfjaldfwwwwwwwwwwadafdw
    kajf;kdsajkflasasfkjafdaff
    akdjf;ldasjkf;safdfdasfdas
    kaj;fdljk98dafdas43adfadfa
    -----END CERTIFICATE-----
    ```

    Если это не так, добавьте или скорректируйте эти строки.

Удаление сертификата:

Иногда возникает необходимость удалить сертификат.

Примечание: Перед удалением сертификата рекомендуется сохранить его резервную копию на случай, если в дальнейшем потребуется возобновить работу с ним.

Для удаления сертификата из базы данных выполните следующие действия:

1. Запустите Диспетчер ключей, если он еще не запущен:

   ```
   # certmgr
   ```

2. В главном окне откройте меню **Файл базы ключей** и выберите пункт **Открыть**.
3. Выберите файл базы ключей, из которого нужно удалить сертификат, и нажмите кнопку **Открыть**.
4. Введите пароль и нажмите кнопку **ОК**. Если пароль указан правильно, то вновь отобразится окно **Работа с ключами IBM**. В заголовке окна будет показано имя выбранного файла базы ключей. Это означает, что файл открыт и готов к работе.
5. Войдите в меню **Сертификаты** и выберите **Запросы на личные сертификаты**.
6. Выберите сертификат и нажмите кнопку **Удалить**. Появится окно **Подтверждение**.

7. Нажмите кнопку **Да**. Вновь отобразится окно **Работа с ключами IBM**. Удаленный сертификат теперь не будет показан в поле **Личные сертификаты**. Теперь можно перейти к выполнению других задач или закончить работу с этой программой.

Изменение пароля для доступа к базе данных:

Время от времени пароль для доступа к базе данных необходимо менять.

Для изменения пароля базы ключей выполните следующие действия:

1. Запустите Диспетчер ключей, если он еще не запущен:
   ```
   # certmgr
   ```
2. В главном окне откройте меню **Файл базы ключей** и выберите пункт **Изменить пароль**.
3. Введите новый пароль в поле **Пароль** и в поле **Подтверждение пароля**.
4. Для изменения числа дней до истечения срока действия введите нужное значение в днях в поле **Задать срок действия?**. По умолчанию срок действия пароля составляет 60 дней. Для задания неограниченного срока действия пароля выключите поле **Задать срок действия?**.
5. Для сохранения зашифрованной версии пароля в бумажнике включите поле **Сохранить пароль в файле?** и выберите Да.

   Примечание: Для применения цифровых сертификатов со средствами защиты IP сохранение пароля в файле обязательно.
6. Нажмите кнопку **ОК**. В строке состояния будет показано сообщение о выполнении операции.
7. Снова нажмите **ОК** для возврата в окно **Работа с ключами IBM**. Теперь можно перейти к выполнению других задач или закончить работу с этой программой.

Создание туннелей IKE, использующих цифровые сертификаты:

Для создания туннелей IKE, в которых применяются цифровые сертификаты, в файле стратегии преобразования туннеля IKE нужно указать, что в качестве режима идентификации используются подписи RSA.

Ниже приведен пример XML-файла стратегии, в котором заданы подписи RSA:

```
<!-- определить стратегию для туннеля IKE -->
<IKEProtection
 IKE ProtectionName="ike_3des_sha">
 <IKETTransform
  IKE AuthenticationMethod="RSA_signatures"
  IKE Encryption="3DES-CBC"
  IKE Hash="SHA"
  IKE DHGroup="1"/>
</IKEProtection>
```

Средства защиты IP поддерживают следующие типы субъектов хостов туннелей IKE:

- IP address
- Полное доменное имя (FQDN)
- *user@FQDN*
- Отличительное имя X.500
- Идентификатор ключа

Когда туннель IKE использует режим подписей RSA, в определении туннелей IKE обычно используются отличительные имена X.500. Например, если локальный и удаленный хосты туннеля идентифицируются как **/C=US/O=ABC/OU=SERV/CN=localname.austin.ibm.com** и **/C=US/O=ABC/OU=SERV/CN=remotename.austin.ibm.com**, то определение туннеля IKE в XML-файле имеет примерно такой вид:

```
<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>
```

Для того чтобы получить требуемый сертификат от сертификатной компании (CA) используйте администратор ключей для создания запроса на сертификат. Например, если в сертификате указано отличительное имя субъекта **/C=US/O=ABC/OU=SERV/CN=name.austin.ibm.com**, то в инструменте Администратор ключей необходимо указать следующие значения во время создания запроса на цифровой сертификат:

Имя *name.austin.ibm.com*

Организация
ABC

Подразделение организации
SERV

Страна
США

Отличительное имя X.500 - это имя, которое обычно задается администратором системы или LDAP. Подразделение указывать необязательно.

Защита IP также поддерживает ввод других типов субъектов в качестве альтернативных имен субъектов в цифровом сертификате. Например, если в качестве альтернативного субъекта хоста используется IP-адрес 10.10.10.1, то в запросе на цифровой сертификат необходимо ввести следующие значения:

Имя *name.austin.ibm.com*

Организация
ABC

Подразделение организации
SERV

Страна
США

Поле альтернативного IP-адреса субъекта
10.10.10.1

Сертификатная компания занесет эти данные в ваш сертификат.

В запросе на сертификат нужно указать следующую информацию для сертификатной компании:

- Вы запрашиваете сертификат X.509.
- Формат сертификата - MD5 с шифрованием RSA.
- Указываете ли вы альтернативное имя. Типы альтернативных имен указаны в следующем списке:
  - IP address
  - Полное доменное имя (FQDN)
  - *user@FQDN*

  В файл запроса на сертификат включается следующая информация об альтернативном имени:

- Предполагаемый способ работы с ключом (должен быть включен режим цифровых подписей).
- Файл запроса на сертификат (в формате PKCS#10).

Процедура использования инструмента Администратор ключей для создания запроса на сертификат описана в разделе “Запрос сертификата” на стр. 245.

Перед активацией туннеля IKE нужно импортировать полученный сертификат в базу данных Диспетчера ключей (`ikekey.kdb`). Дополнительная информация приведена в разделе “Получение и установка сертификата” на стр. 245.

Средства защиты IP поддерживают следующие типы личных цифровых сертификатов:

Отличительное имя владельца
Отличительное имя владельца должно быть указано в следующем формате:

`/C=RU/O=IJK/OU=SERV/CN=`*`name`*`.serv.ijk.ru`

Можно указать не более одного поля **OU**.

DN и альтернативное имя владельца в форме IP-адреса
Отличительное и альтернативное имена владельца можно указать в форме IP-адреса:

`/C=US/O=ABC/OU=SERV/CN=name.austin.ibm.com` и `10.10.10.1`

DN и альтернативное имя владельца в форме FQDN
Отличительное и альтернативное имена владельца можно указать в форме полного доменного имени:

`/C=US/O=ABC/OU=SERV/CN=name.austin.ibm.com` и `bell.austin.ibm.com`.

DN и альтернативное имя владельца в форме *user@FQDN*
Отличительное и альтернативное имена владельцев можно указать в форме почтового адреса пользователя (*пользователь@полное-доменное-имя*):

`/C=US/O=ABC/OU=SERV/CN=name.austin.ibm.com` и `name@austin.ibm.com`.

DN и несколько альтернативных имен владельца
Отличительное имя владельца может быть связано с несколькими альтернативными именами:

`/C=US/O=ABC/OU=SERV/CN=name.austin.ibm.com` и `bell.austin.ibm.com`, `10.10.10.1` и `user@name.austin.ibm.com`.

Преобразование сетевых адресов

Реализация защиты IP поддерживает устройства, адреса которых преобразуются с помощью функции Преобразование сетевых адресов (NAT).

NAT широко применяется как часть технологии брандмауэра для совместного использования соединений Internet и является стандартной функцией маршрутизаторов и оконечных устройств. Работа протокола защиты IP во многом зависит от идентификации удаленных конечных точек и их стратегии по удаленному IP-адресу. Если внутренний адрес преобразуется во внешний адрес промежуточными устройствами, такими как маршрутизаторы и брандмауэры, то во время выполнения применяемой в защите IP идентификации может возникать ошибка, поскольку адрес в пакете IP изменяется после вычисления контрольного значения. Новые средства поддержки NAT в защите IP позволяют устройствам, расположенным за узлом, выполняющим преобразование сетевых адресов, устанавливать туннель защиты IP. Программный код защиты IP может определить, что удаленный адрес был преобразован. Использование новой реализации защиты IP с поддержкой NAT позволяет клиенту VPN подключаться к корпоративной сети из дома или по дороге на работу, используя соединение Internet, применяющее NAT.

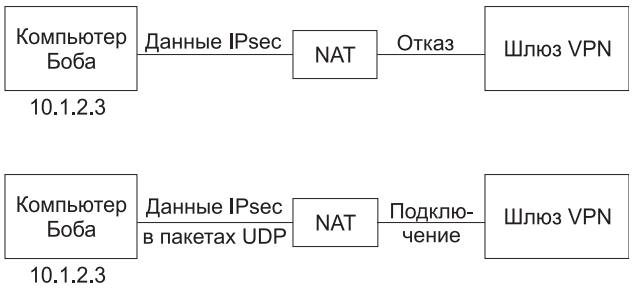


Рисунок 12. Защита IP с поддержкой NAT

На этой схеме показано различие между реализацией защиты IP с поддержкой NAT и реализацией без поддержки NAT на примере потока данных UDP.

Настройка защиты IP для работы с NAT:

Для использования поддержки NAT в защите IP необходимо настроить переменную *ENABLE_IPSEC_NAT_TRAVERSAL* в файле `/etc/isakmpd.conf`. Когда задана эта переменная, добавляются правила фильтрации для отправки и приема данных через порт 4500.

В следующем примере показаны те правила фильтрации, которые добавляются, если задана переменная *ENABLE_IPSEC_NAT_TRAVERSAL*.

```
Динамическое правило 2:
Действие правила    : принять
Адрес отправителя   : 0.0.0.0 (любой)
Маска отправителя   : 0.0.0.0 (любая)
Адрес получателя    : 0.0.0.0 (любой)
Маска получателя    : 0.0.0.0 (любая)
Маршрут отправителя : нет
Протокол            : все
Исходный порт       : 0 (любой)
Порт получателя     : 4500
Принадлежность      : локальная
Направление         : входящие
Фрагментация        : все пакеты
Номер ИД туннеля    : 0

Динамическое правило 3:
Действие правила    : принять
Адрес отправителя   : 0.0.0.0 (любой)
Маска отправителя   : 0.0.0.0 (любая)
Адрес получателя    : 0.0.0.0 (любой)
Маска получателя    : 0.0.0.0 (любая)
Маршрут отправителя : нет
Протокол            : все
Исходный порт       : 4500
Целевой порт        : 0 (любой)
Принадлежность      : локальная
Направление         : исходящие
Фрагментация        : все пакеты
Номер ИД туннеля    : 0
```

При настройке переменной *ENABLE_IPSEC_NAT_TRAVERSAL* некоторые правила фильтрации добавляются и в таблицу фильтров. Специальные сообщения IPSEC NAT инкапсулируются в дейтаграммы UDP, и для разрешения этого потока данных необходимо добавить правила фильтрации. Кроме этого, на этапе 1 необходим режим подписи. Если в сертификате роль идентификатора играет IP-адрес, то сертификат должен содержать внутренний IP-адрес.

Для преобразования исходных адресов IP в адреса NAT в защите IP требуется отправлять контрольные сообщения NAT. Интервал отправки сообщений задается в переменной *NAT_KEEPALIVE_INTERVAL* в файле `/etc/isakmpd.conf`. Эта переменная определяет время в секундах между отправкой контрольных пакетов NAT. Если значение переменной *NAT_KEEPALIVE_INTERVAL* не задано, то используется значение по умолчанию, равное 20 секундам.

Ограничения при использовании замены NAT:

Расположенные за устройствами NAT конечные точки должны применять протокол ESP для защиты передаваемых данных.

ESP чаще всего применяется в защите IP, поэтому он поддерживается в большинстве пользовательских приложений. ESP использует хэширование пользовательских данных, но не заголовка IP. При использовании заголовка AH выполняется проверка целостности данных с использованием адресов отправителя и получателя. Устройства, выполняющее прямое или обратное преобразование NAT, изменяют поля адреса, что приводит к ошибкам при проверке целостности сообщения. Поэтому если на этапе 2 для туннеля определен только протокол AH, и на этапе 1 был обнаружен NAT, то отправляется пакет уведомления `NO_PROPOSAL_CHOSEN`.

Кроме этого, для соединения, использующего NAT, следует выбрать режим туннеля для включения исходного IP-адреса в пакет. Открытый режим несовместим с применением NAT. Если обнаружено преобразование NAT, и на этапе 2 предложен только открытый режим, то отправляется пакет уведомления `NO_PROPOSAL_CHOSEN`.

Предотвращение конфликтов режимов туннеля:

Удаленные равноправные узлы могут согласовать записи, которые пересекаются на шлюзе. Это может вызвать конфликт в режиме туннеля.

На следующей схеме показан пример такого конфликта.

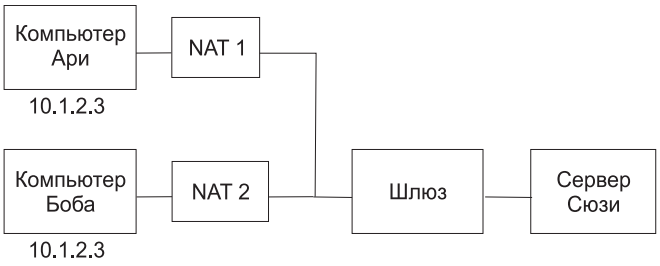


Рисунок 13. Конфликт в режиме туннеля

Шлюз имеет две возможные конфигурации защиты (SA) для IP-адреса 10.1.2.3. Наличие двух одинаковых удаленных адресов вызывает путаницу в том, куда следует отправлять пакеты, поступающие от сервера. При настройке туннеля между сервером Сьюзи и портативным компьютером Ари используется IP-адрес, поэтому Сьюзи не может настроить туннель с Бобом, используя тот же IP-адрес. Для того чтобы избежать возникновения конфликта в режиме туннеля, не следует определять туннели с одним и тем же IP-адресом. Так как удаленный адрес не контролируется удаленным пользователем, для идентификации удаленного хоста следует использовать другие типы ИД, например полное имя домена или имя пользователя с полным именем домена.

Настройка статических туннелей

Если устройства не поддерживают метод автоматической передачи ключей, то можно настроить статические туннели IPSec.

Статические туннели и фильтры:

Процесс настройки туннеля состоит в определении туннеля в одной системе, импорте определения этого туннеля из файла в другой системе и активации туннеля и правил фильтрации в обеих системах. После этого туннель готов к использованию.

Для настройки статического туннеля не требуется отдельно настраивать правила фильтрации. Нужные правила фильтрации создаются автоматически при передаче данных между хостами.

Если информация о туннеле не задана явно, она выбирается одинаковой на обоих концах. Например, алгоритмы шифрования и идентификации, заданные в одной системе, будут использоваться в другой системе, если они не были указаны там явно.

Создание статического туннеля в первой системе:

Настроить туннель можно с помощью команды быстрого доступа SMIT `ips4_basic` (для IP версии 4) или `ips6_basic` (для IP версии 6). Кроме того, можно создать туннель вручную с помощью следующей процедуры.

Ниже приведен пример применения команды **gentun** для создания статического туннеля:

```
gentun -v 4 -t manual -s 5.5.5.19 -d 5.5.5.8 \
   -a HMAC_MD5 -e DES_CBC_8 -N 23567
```

Характеристики статического туннеля, созданного в предыдущем примере, можно просмотреть командой **lstun -v 4**. Ниже приведен пример вывода этой команды:

```
ИД туннеля           : 1
Версия протокола IP  : IP версии 4
Источник             : 5.5.5.19
Назначение           : 5.5.5.8
Стратегия            : идентификация/шифрование
Режим туннеля        : Туннель
Алгоритм отправки AH : HMAC_MD5
Алгоритм отправки ESP: DES_CBC_8
Алгоритм приема AH   : HMAC_MD5
Алгоритм приема ESP  : DES_CBC_8
SPI AH источника     : 300
SPI ESP источника    : 300
SPI AH назначения    : 23576
SPI ESP назначения   : 23576
Срок действия туннеля: 480
Состояние            : Неактивен
Назначение           : -
Маска назначения     : -
Повтор               : Нет
Новый заголовок      : Да
Алг. отправки ENC-MAC: -
Алг. приема ENC-MAC  : -
```

Для активации туннеля введите следующую команду:

```
mktun -v 4 -t1
```

Правила фильтрации, связанные с туннелем, будут созданы автоматически.

Для просмотра правил фильтрации введите команду **lsfilt -v 4**. Ниже приведен пример вывода этой команды:

```
Правило 4:
Действие правила    : принять
Адрес источника       : 5.5.5.19
Маска источника       : 255.255.255.255
Адрес назначения      : 5.5.5.8
Маска назначения      : 255.255.255.255
```

```
Маршрут. источника    : да
Протокол              : все
Порт источника        : любой 0
Порт назначения       : любой 0
Принадлежность       : взаимная
Направление          : исходящие
Регистрация          : нет
Фрагментация         : все пакеты
Номер ИД туннеля      : 1
Интерфейс            : все
Автоматическое        : да

Правило 5:
Действие правила    : принять
Адрес источника       : 5.5.5.8
Маска источника       : 255.255.255.255
Адрес назначения      : 5.5.5.19
Маска назначения      : 255.255.255.255
Маршрут. источника    : да
Протокол              : все
Порт источника        : любой 0
Порт назначения       : любой 0
Принадлежность       : взаимная
Направление         : входящие
Регистрация          : нет
Фрагментация         : все пакеты
Номер ИД туннеля      : 1
Интерфейс            : все
Автоматическое        : да
```

Для активации правил фильтрации, в том числе правил по умолчанию, введите команду **mktun -v 4 -t 1**.

Для настройки удаленной системы (если в ней применяется та же операционная система) определение туннеля можно экспортировать в системе A, и затем импортировать в системе B.

Следующая команда экспортирует определение туннеля в файл с именем **ipsec_tun_manu.exp**, а все связанные правила фильтрации - в файл с именем **ipsec_fltr_rule.exp** в каталог, заданный флагом **-f**:

```
exptun -v 4 -t 1 -f /tmp
```

Создание статического туннеля во второй системе:

Для создания второго конца туннеля экспортированные файлы необходимо скопировать в другую систему.

Для создания второго конца туннеля служит команда:

```
imptun -v 4 -t 1 -f /tmp
```

где

1 Импортируемый туннель

/tmp Каталог, в котором находятся импортируемые файлы

Номер туннеля создается системой. Его можно просмотреть в выводе команды **gentun** или **lstun** с флагами просмотра списка туннелей. Если файл импорта содержит только один туннель, или необходимо импортировать все находящиеся в нем туннели, опция **-t** не обязательна.

Если удаленная система работает под управлением другой операционной системы, файл экспорта можно использовать для справки при задании алгоритма, ключей и SPI.

При создании туннеля можно использовать файлы, экспортированные брандмауэром. Для этого укажите при импорте опцию **-n**:

```
imptun -v 4 -f /tmp -n
```

Удаление фильтров:

Для удаления фильтров и отключения защиты IP используйте команду **rmdev**.

Правило фильтра по умолчанию остается активным, даже если фильтрация была отключена командой **mkfilt -d**. С помощью этой команды можно приостановить или удалить все правила фильтров, загрузить новые правила фильтров, пока правило фильтра по умолчанию обеспечивает защиту. Правило фильтра по умолчанию - это правило *DENY*. Если фильтрация деактивирована командой **mkfilt -d**, в отчетах, создаваемых командой **lsfilt** будет указано, что фильтрация выключена, но отправка и получение пакетов будет запрещено. Для того, чтобы полностью отключить защиту IP, используйте команду **rmdev**.

Настройка фильтра защиты IP

Фильтр может быть простым, когда он основан преимущественно на правилах, созданных автоматически, или сложным, когда используются разнообразные правила, созданные пользователем с учетом свойств передаваемых IP-пакетов.

Каждая строка в таблице фильтрации называется *правилом*. Набор правил определяет, какие из получаемых и отправляемых пакетов принимаются, а какие - отклоняются. Проверка входящих пакетов по правилам фильтрации выполняется путем сравнения адреса источника и значения SPI со значениями, указанными в таблице фильтрации. Таким образом, данная пара должна быть уникальной. Правила фильтрации могут управлять многими параметрами связи, включая адреса и маски источника и получателя, протокол, номер порта, направление, фрагментацию, маршрутизацию источника, туннель и тип интерфейса.

Ниже перечислены типы правил фильтрации:

- Правила статических фильтров создаются для общей фильтрации пакетов, либо для статических туннелей. Их можно добавлять, удалять, изменять и перемещать. К каждому правилу можно добавить необязательный текст описания.
- Правила фильтров, созданных автоматически и правила фильтров, созданных пользователем (также называемые *автоматическими* правилами фильтрации) - создаются для работы туннелей IKE. Как статические, так и динамические правила фильтрации создаются на основе информации туннеля управления данными при согласовании этой информации.
- Стандартные правила фильтров общие правила фильтрации, которые нельзя изменять, перемещать и удалять; например, правило для `всего потока`, правило `ah`, правило `esp`. Эти правила относятся ко всему потоку данных.

Метка направления (**-w**) для команды **genfilt** указывает, используется ли правило для обработки входящих или исходящих пакетов. Если для этой метки используется значение **both**, это означает, что правило используется для обработки и входящих, и исходящих пакетов. В AIX IPsec, при включенной фильтрации, как минимум одно правило определяет судьбу любого сетевого пакета (независимо от того, входящий он или исходящий). Если вы хотите, чтобы правило применялось только при обработке исходящих (или входящих) пакетов, используйте переключатель **-w** команды **genfilt**. Например, если пакет отправляется из хоста A хосту B, то исходящий пакет IP имеет исходный адрес *A* и целевой адрес *B*. Этот пакет обрабатывается фильтром IPsec, как исходящий пакет на хосте A, и как входящий на хосте B. Предположим, между хостом A и B есть шлюз G. На шлюзе G этот же пакет (постоянные поля с теми же значениями) обрабатываются дважды: как входящий и как исходящий пакет (если включена опция **ipforwarding**). Для пакета, отправляемого с хоста A на хост B через шлюз G, необходимо разрешающее правило:

- На хосте A – **src addr** заданное для `A`, **dest addr** для `B`, направление на отправку
- На хосте B – **src addr** заданное для `A`, **dest addr** для `B`, направление на получение

Для шлюза G требуются для правила:

1. **src addr** заданное для `A`, **dest addr** для `B`, направление на отправку
2. **src addr** заданное для `A`, **dest addr** для `B`, направление на получение

Вышеприведенное правило можно заменить на: **src addr** заданное дляA, **dest addr** для B и направление both (и на оправку, и на получение). Поэтому значение **both** для направления обычно используется в шлюзах, у которых параметр **ipforwarding** имеет значение no. Настройки, приведенные выше, предназначены для пакетов, отправленных с хоста A на хост B через шлюз G. Для пакетов, следующих в обратном направлении (с хоста B на хост A через шлюз G), необходимо другое правило.

Примечание: При использовании направления **both** предполагается, что соответствующее правило используется как для входящих, так и для исходящих пакетов. Однако, это не означает, что правило будет применяться, если адреса отправителя и получателя поменять местами. Например, если на сервере A задано правило с адресом отправителя *A* и адресом получателя *B*, а для направления задано **both**, то входящий пакет для *A* с адресом отправителя *B* и адресом получателя *A* не будет соответствовать правилу. Опция **both** обычно используется для шлюзов, которые пересылают пакеты.

С этими правилами фильтрации связаны маски подсети, идентификаторы группы и опции конфигурации хост-брандмауэр-хост. Следующие сценарии описывают различные типы правил фильтрации и их свойства.

Фильтры IP для AIX:

Пакет IPFilter позволяет воспользоваться службами брандмауэра и преобразования сетевых адресов (NAT).

Программное обеспечение с открытым исходным кодом IPFilter версии 4.1.13 было перенесено на AIX, соответствующую лицензии, которая представлена на Web-сайте IP Filter (http://coombs.anu.edu.au/~avalon/). Программное обеспечение IPFilter поставляется в составе пакета расширения AIX. В пакет установки ipfl включены страница справки и лицензия.

IPFilter в операционной системе AIX загружается как расширение ядра /usr/lib/drivers/ipf. Двоичные файлы **ipf**, **ipfs**, **ipfstat**, **ipmon** и **ipnat** также поставляются в этом пакете.

После установки пакета выполните следующую команду, чтобы загрузить расширение ядра:

```
/usr/lib/methods/cfg_ipf -l
```

Для того чтобы выгрузить расширение ядра выполните следующую команду:

```
/usr/lib/methods/cfg_ipf -u
```

Если необходимо перенаправить пакет, не забудьте включить опцию ipforwarding (опция сети). Дополнительные сведения об IPFilter, включая справочные материалы и Часто задаваемые вопросы, приведены на Web-сайте IPFilter (http://coombs.anu.edu.au/~avalon/).

Статические правила фильтрации:

Каждое статическое правило фильтрации состоит из нескольких разделенных пробелами полей.

Названия полей перечислены в следующем списке (в скобках показан пример значения для правила 1):

- Rule_number - номер правила (1)
- Action - действие (permit - принять)
- Source_addr - адрес источника (0.0.0.0)
- Source_mask - маска источника (0.0.0.0)
- Dest_addr - адрес получателя (0.0.0.0)
- Dest_mask - маска получателя (0.0.0.0)
- Source_routing - маршрутизация источника (no - нет)
- Protocol - протокол (udp)
- Src_prt_operator - оператор порта источника (eq - сравнение)
- Src_prt_value - значение порта источника (4001)

- Dst_prt_operator - оператор порта получателя (eq - сравнение)
- Dst_prt_value - значение порта получателя (4001)
- Scope - принадлежность (both - взаимная)
- Direction - направление (both - оба)
- Logging - регистрация (no - нет)
- Fragment - фрагментация (all packets - все пакеты)
- Tunnel - туннель (0)
- Interface - интерфейс (all - все).

Пример статических правил фильтрации

```
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 исходящие данные

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
```

Описание перечисленных правил фильтрации:

Правило 1

Предназначено для демона **ключей сеанса**. Это правило добавляется только в таблицы фильтрации IP версии 4. Оно использует порт 4001 для управления пакетами обновления ключа сеанса. Правило 1 демонстрирует, каким образом можно указать в правиле номер порта.

Примечание: Не изменяйте данное правило фильтрации, кроме как для включения регистрации.

Правила 2 и 3

Разрешают обработку заголовков AH и ESP.

Примечание: Не изменяйте правила 2 и 3, кроме как для включения регистрации.

Правила 4 и 5

Автоматические правила, применяемые для фильтрации потока данных между адресами 10.0.0.1 и 10.0.0.2 через туннель 1. Правило 4 применяется к исходящему потоку, правило 5 - ко входящему.

Примечание: С правилом 4 связано пользовательское описание *исходящие данные*.

Правила 6-9

Набор пользовательских правил, применяемых для фильтрации исходящих потоков данных служб rsh, rcp, rdump, rrestore и rdist между адресами 10.0.0.1 и 10.0.0.3 через туннель 2. В этом примере в поле регистрации указано yes (да), поэтому администратор может отслеживать данные соответствующего типа.

Правила 10 и 11

Пользовательские правила, применяемые для фильтрации исходящего и входящего потока данных служб icmp любого типа между адресами 10.0.0.1 и 10.0.0.4 через туннель 3.

Правила с 12 по 17

Пользовательские правила, применяемые для фильтрации исходящего потока данных службы FTP между адресами 10.0.0.1 и 10.0.0.5 через туннель 4.

Правило 18

Автоматическое правило, всегда помещаемое в конец таблицы. В данном примере это правило указывает, что все пакеты, не соответствующие остальным правилам фильтрации, принимаются. Можно изменить его так, чтобы все пакеты, не соответствующие остальным правилам фильтрации, отклонялись.

Каждое правило можно просмотреть отдельно (командой **lsfilt**); при этом будет показано название и значение каждого поля правила. Например:

```
Правило 1:
Действие правила    : принять
Адрес источника      : 0.0.0.0
Маска источника      : 0.0.0.0
Адрес получателя     : 0.0.0.0
Маска получателя     : 0.0.0.0
```

```
Маршрут. источника   : да
Протокол            : все
Порт источника       : eq  4001
Порт получателя      : eq  4001
Принадлежность       : взаимная
Направление          : оба
Регистрация          : нет
Фрагментация         : все пакеты
ИД туннеля           : 0
Интерфейс            : все
Автоматическое       : да
```

Все параметры, которые могут быть указаны в правиле фильтрации, перечислены в следующем списке:

-v Версия IP: 4 или 6.

-a Действие:

- **d** Отклонить
- **p** Принять

-s Адрес источника. IP-адрес или имя хоста.

-m Маска подсети источника.

-d Адрес получателя. IP-адрес или имя хоста.

-M Маска получателя.

-g Управление маршрутизацией источника: `y` (да) или `n` (нет).

-c Протокол. Допустимые значения: `udp`, `icmp`, `tcp`, `tcp/ack`, `ospf`, `pip`, `esp`, `ah` и `all`.

-o Порт источника или тип операции ICMP.

-p Порт источника или значение типа ICMP.

-O Порт получателя или операция кода ICMP.

-P Порт получателя или значение кода ICMP.

-r Маршрутизация:

- **r** Переданные пакеты.
- **l** Локальные входящие/исходящие пакеты.
- **b** Оба типа.

-l Режим регистрации.

- **y** Включить в протокол.
- **n** Не включать в протокол.

-f Фрагментация.

- **y** Применяется к заголовкам фрагментов, фрагментам и нефрагментированным пакетам.
- **o** Применяется только к фрагментам и заголовкам фрагментов.
- **n** Применяется только к нефрагментированным пакетам.
- **h** Применяется только к нефрагментированным пакетам и заголовкам фрагментов.

-t Идентификатор туннеля.

-i Интерфейс; например, `tr0` или `en0`.

Дополнительная информация приведена в описаниях команд **genfilt** и **chfilt**.

Автоматические и пользовательские правила фильтрации:

Некоторые правила, предназначенные для защиты IP-пакетов и работы туннелей, создаются автоматически.

К автоматическим правилам относятся следующие наборы правил:

- Правила для демона ключей сеанса, обновляющего ключи IP4 в IKE
- Правила обработки пакетов AH и ESP.

Кроме того, правила фильтрации автоматически создаются при определении туннелей. Для статических туннелей автоматические правила содержат адреса и маски источника и получателя, а также идентификатор туннеля. Весь поток данных между указанными адресами передается через туннель.

Для туннелей IKE автоматические правила фильтрации, задают протокол и номер порта при согласовании IKE. Правила фильтрации IKE хранятся в отдельной таблице, просматриваемой после статических правил фильтрации, но перед автоматическими правилами. Правила фильтрации IKE вставляются в расположение по умолчанию в статической таблице фильтрации, но могут быть перемещены пользователем.

Автоматические правила пропускают весь поток данных, передаваемых через туннель. Пользовательские правила могут налагать ограничения на некоторые типы потоков данных. Такие правила следует размещать перед автоматическими, поскольку защита IP-пакетов применяет к каждому пакету первое подходящее правило в списке. Ниже приведен пример пользовательских правил, фильтрующих поток данных на основе операции 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
```

Для того чтобы упростить настройку туннелей, правила фильтрации создаются автоматически при определении туннелей. Эту функцию можно отключить, указав в команде **gentun** флаг **-g**. Примеры правил фильтрации и соответствующих команд **genfilt** для различных служб TCP/IP приведены в файле `/usr/samples/ipsec/filter.sample`.

Предопределенные правила фильтрации:

Несколько предопределенных правил фильтрации автоматически создаются при определенных событиях.

Например, предопределенное правило вставляется в таблицу фильтрации и активируется при загрузке устройства `ipsec_v4` или `ipsec_v6`. По умолчанию это предопределенное правило принимает все пакеты, но его можно изменить так, чтобы все пакеты отклонялись.

Примечание: При удаленной настройке не включайте отклоняющее правило, пока не завершите настройку, во избежание отключения сеанса настройки от удаленной системы. Для этого либо оставьте правило по умолчанию принимающим все пакеты, либо активируйте защиту IP-пакетов в удаленной системе после завершения настройки.

Предопределенные правила для таблиц фильтрации есть как для IP версии 4, так и для IP версии 6. Каждое из них может быть превращено в правило, отклоняющее все пакеты. Это предотвратит передачу данных, не разрешенных специальными правилами фильтрации. Единственная дополнительная возможность изменения предопределенных правил - команда **chfilt** с опцией **-l**, включающая регистрацию пакетов, соответствующих этому правилу.

Для обеспечения поддержки туннелей IKE динамическое правило фильтрации помещается в таблицу фильтрации IP4. Оно вставляется в ту часть таблицы, где располагаются динамические правила фильтрации.

Пользователь может изменить расположение этого правила путем перемещения его вверх или вниз по таблице. После инициализации демона администратора туннелей и демона **isakmpd**, необходимых для работы туннелей IKE, в таблице фильтрации автоматически создаются правила, принимающие сообщения IKE, а также пакеты AH и ESP.

Маски подсети:

Маски подсети применяются для объединения в группу набора идентификаторов, связанных с одним правилом фильтрации. Двоичное значение маски логически умножается на идентификатор в правилах фильтрации и сравнивается с идентификатором пакета.

Например, правило фильтрации с IP-адресом источника `10.10.10.4` и маской подсети `255.255.255.255` указывает, что требуется точное совпадение десятичного IP-адреса, как показано ниже:

	Двоичный вид	**Десятичный вид**
IP-адрес источника	`1010.1010.1010.0100`	`10.10.10.4`
Маска подсети	`11111111.11111111.11111111.11111111`	`255.255.255.255`

Подсеть `10.10.10.x` обозначается маской `11111111.11111111.11111111.0` или `255.255.255.0`. Маска подсети будет применена к адресу источника, после чего результат будет сравниваться с идентификатором в правиле фильтрации. Например, адрес `10.10.10.100` после применения маски подсети будет преобразован в `10.10.10.0`, что соответствует правилу фильтрации.

Маска подсети `255.255.255.240` допускает произвольные значения в четырех последних разрядах адреса.

Конфигурация хост-брандмауэр-хост:

Опция конфигурации хост-брандмауэр-хост позволяет создать туннель между хостом и брандмауэром, а затем автоматически создать необходимые правила фильтрации для управления связью между этим хостом и хостом, защищенным брандмауэром.

Автоматические правила фильтрации принимают все пакеты, которые передаются между двумя этими хостами через указанный туннель. Правила по умолчанию для протоколов UDP, AH и ESP автоматически пропускают большую часть потока данных между хостом и брандмауэром. Для завершения настройки требуется соответствующая настройка брандмауэра. Для ввода значений SPI и ключей, необходимых брандмауэру, рекомендуется воспользоваться экспортированным файлом.

На этом рисунке показана конфигурация хост-брандмауэр-хост. Туннель идет от хоста A через локальный

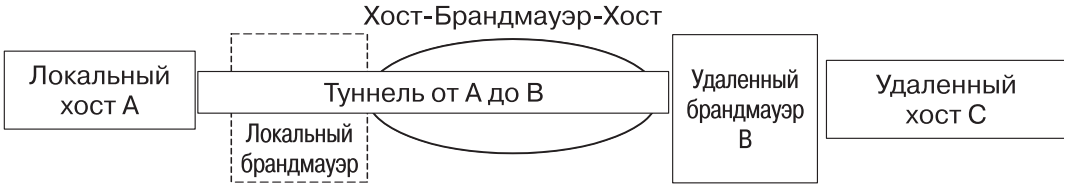


Рисунок 14. Хост-брандмауэр-хост

брандмауэр в Internet. Затем он проходит через удаленный брандмауэр B к хосту C.

Средства ведения протокола

Пакеты, которыми обмениваются хосты, могут заноситься в протокол демоном `syslogd`. Кроме того, в протокол заносятся и прочие сообщения системы защиты IP.

Администратор может указать, какая именно информация должна сохраняться в протоколе. Ниже приведена процедура настройки средств ведения протокола.

1. Добавьте в файл /etc/syslog.conf следующую строку:

   ```
   local4.debug var/adm/ipsec.log
   ```

   Для протоколирования событий защиты IP применяется функция local4. Используется стандартная система приоритетов операционной системы. Пока работа туннелей IP не стабилизируется, рекомендуется применять приоритет debug.

   Примечание: На ведение протокола могут потребоваться значительные вычислительные ресурсы и много дисковой памяти.
2. Сохраните файл /etc/syslog.conf.
3. Перейдите в каталог, в котором хранится файл протокола, и создайте пустой файл с таким же именем. В данном случае нужно перейти в каталог /var/adm и выполнить следующую команду:

   ```
   touch ipsec.log
   ```
4. Выполните команду **refresh** для подсистемы syslogd:

   ```
   refresh -s syslogd
   ```
5. При работе с туннелями IKE степень детализации протокола **isakmpd** задается в файле /etc/isakmpd.conf. (Дополнительная информация о протоколах IKE приведена в разделе “Диагностика неполадок защиты протокола IP” на стр. 266.)
6. Если требуется, чтобы в протокол заносились сведения по определенному правилу фильтрации, созданному для хоста, то укажите значение параметра *-l* для этого правила равным **Y** (Yes) с помощью команды **genfilt** или **chfilt**.
7. Включите режим ведения протокола и запустите демон **ipsec_logd** с помощью следующей команды:

   ```
   mkfilt -g start
   ```

   Ведение протокола можно прекратить с помощью следующей команды:

   ```
   mkfilt -g stop
   ```

Следующий файл содержит примеры записей протокола, связанных с передачей данных и защитой IP:

```
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
```

Описание этих записей приведено ниже.

1 Активирован демон протокола фильтров.

2 Протокол включен командой **mkfilt -g start**.

3 Активирован туннель с указанными ИД, исходным и целевым адресом и меткой времени.

4-9 Активация фильтров. В протоколе показаны все загруженные правила фильтрации.

10 Сообщение об активации фильтров.

11-12 Результаты поиска хостов в DNS.

13-15 Часть записей, относящихся к соединению Telnet (остальные записи удалены для экономии места).

16-19 Результаты выполнения двух команд ping.

20 Демон протокола фильтров выключен.

В следующем примере показан процесс согласования туннелей первого и второго этапов со стороны инициатора. (Для **isakmpd** установлен режим подробности **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
```

Описание этих записей приведено ниже.

1-2 Команда **ike cmd=activate phase=1** устанавливает соединение.

3-10 Демон **isakmpd** согласует туннель первого этапа.

11-12 Диспетчер туннелей получает конфигурацию защиты для этапа 1 от противоположной стороны.

13 Диспетчер туннеля проверяет, присвоено ли параметру **ike cmd=activate** значение 2 (в этом случае потребовались бы дополнительные операции). Однако это не так.

14-16 Демон **isakmpd** завершает согласование первого этапа.

17-21 Команда **ike cmd=activate phase=2** создает туннель второго этапа.

22-29 Демон **isakmpd** согласует туннель второго этапа.

30-31 Диспетчер туннелей получает конфигурацию защиты для этапа 2 от противоположной стороны.

32 Диспетчер туннелей записывает динамические правила фильтрации.

33 Просмотр списка туннелей с помощью команды **ike cmd=list**.

Метки в полях записей протокола:

Сведения в записях протокола приводятся в укороченной форме для выполнения требований DASD к экономии дисковой памяти.

Поле	Описание
#	Номер правила, в результате применения которого сведения о данном пакете были занесены в протокол.
R	Тип правила **p** Разрешить **d** Отклонить
i/o	Направление передачи пакета в момент перехвата. IP-адрес адаптера, связанного с пакетом: • Для входящих (i) пакетов - адаптер, получивший пакет. • Для исходящих (o) пакетов - адаптер, которому протоколом IP была назначена передача пакета.
s	IP-адрес отправителя пакета (из заголовка IP).
d	IP-адрес получателя пакета (из заголовка IP).
p	Протокол высокого уровня, применявшийся для создания сообщения в области данных пакета. Значение может быть числом или именем, например: `udp`, `icmp`, `tcp`, `tcp/ack`, `ospf`, `pip`, `esp`, `ah` или `all`.
sp/t	Номер порта протокола отправителя пакета (из заголовка TCP/UDP). Для протоколов ICMP и OSPF в этом поле указывается значение **t**, указывающее на тип IP.
dp/c	Номер порта протокола получателя пакета (из заголовка TCP/UDP). Для протокола ICMP в этом поле указывается значение **c**, указывающее на тип IP.
-	Говорит о том, что информации нет.

Поле	Описание
r	Указывает, что имел ли пакет непосредственное отношение к локальному хосту.
	f Пересылавшиеся пакеты
	l Локальный
	o Исходящий
	b И то, и другое
l	Длина пакета в байтах.
f	Указывает, является ли пакет фрагментом другого пакета.
T	ИД туннеля.
i	Интерфейс, с которого получен пакет.

Ведение протоколов обмена Internet-ключами:

Можно включить ведение протоколов обмена Internet-ключами в утилиту SYSLOG с помощью демона **isakmpd**.

Для включение ведения протокола демоном **isakmpd** применяется команда **ike cmd=log**. Можно настроить степень детализации протокола в файле конфигурации `/etc/isakmpd.conf` с помощью параметра **`log_level`**. В зависимости от объема информации, которую требуется вносить в протокол, выберите одну из степеней детализации: *none (нет)*, *errors (ошибки)*, *isakmp_events (события isakmp)* или *information (информация)*.

Например, если требуется отслеживать информацию протокола и реализации, укажите параметр следующим образом:

```
log_level=INFORMATION
```

Демон **isakmpd** запускает один из двух процессов: либо отправляет предложение, либо рассчитывает его. Если предложение принято, то создается конфигурация защиты и настраивается туннель. Если предложение не принято или соединение разрывается прежде, чем завершилось согласование, то демон **isakmpd** возвращает сообщение об ошибке. Узнать, было ли согласование успешным, можно по записям **tmd** в файле SYSLOG. Ошибка, вызванная недопустимым сертификатом, также заносится в SYSLOG. Для определения точной причины неудачи при согласовании посмотрите файл протокола, указанный в `/etc/syslog.conf`.

Утилита SYSLOG добавляет к каждой строке файла протокола префикс, в котором указывается дата, время, имя системы и имя программы. В следующем примере указано имя системы `googly` и программа `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
```

Для извлечения из протокола только интересующих записей можно воспользоваться командой**grep** (например, необходимы только записи **isakmpd**), а удалить префикс из каждой строки можно командой **cut**.

Файл `/etc/isakmpd.conf`:

Опции для демона **isakmpd** можно настроить в файле `/etc/isakmpd.conf`.

В файле `/etc/isakmpd.conf` доступны следующие опции.

Конфигурацию протокола

Объем информации, которую требуется вносить в протокол. После этого устанавливается уровень. Демоны IKE по этому параметру определяют степень детализации протокола.

Синтаксис: `none | error | isakmp_events | information`

где уровень может означать следующее:

none Протокол не ведется. Это значение по умолчанию.

error Регистрируются ошибки протокола и ошибки программного интерфейса приложения (API).

isakmp_events
Регистрируются события и ошибки протокола IKE. Этот уровень используется при отладке.

information
Регистрируется информация протокола и информация о реализации.

Согласование с нераспознанным IP-адресом
Этот параметр имеет два значения: "YES" и "NO". При указании значения "YES" локальная база данных IKE должна содержать IP-адрес для обеих конечных точек туннеля этапа 1. Укажите YES, чтобы хост принимал входящий туннель в основном режиме. IP- адрес может быть первичным ИД или необязательным IP-адресом, связанным с ИД другого типа.

Для подтверждения исходящих соединений в основном режиме укажите значение NO. В этом случае хост может принимать соединение даже если в базе данных IKE нет IP-адресов для конечных точек туннеля этапа 1. Однако для того чтобы хост принимал соединение, необходимо применять идентификацию по сертификатам. Она позволит хосту с динамически присваиваемым IP-адресом инициировать туннель в систему в основном режиме.

Если этот параметр не указан, по умолчанию применяется значение NO.

Синтаксис: MAIN_MODE_REQUIRES_IP= YES | NO

Конфигурация сервера SOCKS4
Опция `SOCKS4_PORTNUM` является необязательной. Если она не указана, то значение порта сервера SOCKS по умолчанию равно 1080. Порт нужен при соединении сервера SOCKS с сервером HTTP.

Синтаксис: *мнемоника* = *значение*,
где *мнемоника* и *значение* могут быть следующими:

`SOCKS4_SERVER=` - имя сервера

`SOCKS4_PORTNUM=` номер порта сервера SOCKS

`SOCKS4_USERID=` ИД пользователя

Конфигурация сервера LDAP

Синтаксис: *мнемоника* = *значение*,
где *мнемоника* и *значение* могут быть следующими:

`LDAP_SERVER=` имя сервера LDAP

`LDAP_VERSION=` версия сервера LDAP (2 или 3)

`LDAP_SERVERPORT=` номер порта сервера LDAP

`LDAP_SEARCHTIME=`значение тайм-аута поиска клиента

Порядок выбора CRL
Этот параметр определяет очередность опроса серверов HTTP и LDAP, в случае если настроены оба. Параметр `CRL_FETCH_ORDER` необязателен. По умолчанию, если настроены оба сервера - и HTTP, и LDAP, то первым выбирается сервер HTTP, затем LDAP.

Синтаксис: CRL_FETCH_ORDER= *протокол#*, *протокол#*,
где *протокол#* может принимать значения `HTTP` или `LDAP`.

Спецификация портов IKEv1 и IKEv2.
Эта строка содержит порты, используемые демонами **isakmpd** (IKEv1) и **ikev2d** (IKEv2). Демон **iked** (демон-посредник сообщений IKE) ищет эту строку и запускает демоны **isakmpd** и **ikev2d** на этих портах.

Синтаксис: v1=port-natport,v2=port-natport

Диагностика неполадок защиты протокола IP

В этом разделе приведены советы и рекомендации, которые помогут вам при выявлении и устранении неполадок.

При настройке IPSec в первую очередь настраивается ведение протоколов. Протоколы весьма полезны при анализе работы фильтров и туннелей. (Подробные сведения о протоколах приведены в “Средства ведения протокола” на стр. 260.)

Чтобы узнать, какие демоны защиты IP работают, выполните следующую команду:

```
ps  -ef
```

Демоны, связанные с защитой IP: **tmd**, **iked**, **isakmpd**, **ikev2d**, **cpsd**.

Примечание: Если настроены IKEv1 и IKEv2, выполняется демон **iked**. В остальных случаях, работает демон **iskmpd** или **ikev2d**. Это задается в файле **/etc/isakmpd.conf**.

Устранение неполадок статического туннеля:

В этом разделе описываются возможные ошибки статических туннелей и способы их исправления.

Ошибка	**Возможные неполадки и их решения**
При вводе команды **mktun** выдается следующее сообщение: `insert_tun_man4(): ошибка записи: Запрошенный ресурс занят.`	Неполадка: Туннель, активацию которого вы запросили, уже активен, либо имеет конфликтующие значения SPI. Исправление: Введите команду **rmtun** для деактивации, а затем команду mktun для активации туннеля. Проверьте, соответствуют ли значения SPI активируемого туннеля какому либо другому, уже активному туннелю. Каждому туннелю должен соответствовать уникальный набор значений SPI.

При вводе команды **mktun** выдается следующее сообщение:
 	
`Устройство ipsec_v4 находится в состоянии Определено.`
 	
`Активация туннеля IP версии 4 не выполнена.`
	Неполадка: Вы не перевели устройство защиты IP в доступное состояние.
	
	Исправление: Введите следующую команду:
	`mkdev -l ipsec -t 4`
	
	Если такая ошибка возникает при активации туннеля IP версии 6, то для опции **-t** можно указать значение 6. Устройство должно находиться в доступном состоянии. Для проверки состояния устройства защиты IP введите следующую команду:
	`lsdev -Cc ipsec`

При вводе команды **gentun** выдается следующее сообщение: `Недопустимый исходный IP-адрес.`	Неполадка: Вы не указали допустимый исходный IP-адрес. Исправление: Для туннелей IP версии 4 убедитесь, что указан допустимый IP-адрес версии 4 для локальной системы. В качестве исходной точки туннеля нельзя указывать имя хоста; имя хоста допустимо только в качестве пункта назначения туннеля. Для туннелей IP версии 6 убедитесь, что указан допустимый IP-адрес версии. Если при вводе команды `netstat -in` не показываются IP-адреса версии 6, то запустите `/usr/sbin/autoconf6` (интерфейс) для автоматического присвоения адреса (на основе адреса MAC) или присвойте адрес вручную с помощью команды **ifconfig**.
При вводе команды **gentun** выдается следующее сообщение: `Недопустимый исходный IP-адрес.`	Неполадка: Вы не указали допустимый исходный IP-адрес. Исправление: Для туннелей IP версии 4 убедитесь, что указан допустимый IP-адрес версии 4 для локальной системы. В качестве исходной точки туннеля нельзя указывать имя хоста; имя хоста допустимо только в качестве пункта назначения туннеля. Для туннелей IP версии 6 убедитесь, что указан допустимый IP-адрес версии. Если при вводе команды `netstat -in` не показываются IP-адреса версии 6, то запустите `/usr/sbin/autoconf6` (интерфейс) для автоматического присвоения адреса (на основе адреса MAC) или присвойте адрес вручную с помощью команды **ifconfig**.

Ошибка	Возможные неполадки и их решения
При вводе команды **mktun** выдается следующее сообщение: `insert_tun_man4(): ошибка записи: Системный вызов получил недопустимый параметр.`	Неполадка: При создании туннеля применялось недопустимое сочетание ESP и AH, либо не применялся обязательный новый формат заголовка. Исправление: Проверьте, какие алгоритмы идентификации применяются туннелем. Помните, что алгоритмы HMAC_MD5 и HMAC_SHA требуют применения нового формата заголовков. Новый формат заголовков можно изменить с помощью команды SMIT **ips4_basic** или с помощью параметра *-z* в команде **chtun**. Следует также помнить, что DES_CBC_4 не может применяться с новым форматом заголовка.
При попытке обратиться к защите IP выдается следующее сообщение об ошибке: `Необходимо обновить установленный устаревший набор файлов bos.crypto.`	Неполадка: Файлы `bos.net.ipsec.*` были обновлены, но соответствующие новые версии файлов `bos.crypto.*` не установлены. Исправление: Обновите файлы `bos.crypto.*` до версии, соответствующей обновленным файлам `bos.net.ipsec.*`

Устранение неполадок туннелей обмена Internet-ключами (IKE):

В этом разделе описаны ошибки, которые могут возникать при работе с туннелями IKE.

Как работает туннель IKE:

В этом разделе описывается алгоритм работы туннеля обмена Internet-ключами (IKE).

Туннели IKE настраиваются с помощью команды **ike**, при этом используются следующие демоны:

tmd Демон администратора туннелей.

iked Демон-посредник IKE (активен, только когда в системе настроены демоны IKEv1 и IKEv2).

isakmpd
Демон IKEv1.

ikev2d Демон IKEv2.

cpsd Демон Proxy сертификатов.

Для правильной настройки туннелей IKE демоны **tmd** и **isakmpd** должны быть запущены. Если защита IP запускается при загрузке, то эти демоны запускаются автоматически. В противном случае их следует запускать с помощью следующей команды:

```
startsrc -g ike
```

Для запуска туннеля администратор туннелей передает запрос команде **isakmpd**. Если туннель уже существует или недопустим (например, для него задан недопустимый удаленный адрес), то выдается сообщение об ошибке. Если запущено согласование, то для его выполнения может потребоваться некоторое время, зависящее от скорости сети. В случае успешного согласования определить состояние туннеля можно с помощью команды **ike cmd=list**. События администратора туннелей передаются `syslog` на уровне debug, event и information, и могут применяться для мониторинга процесса согласования.

При этом выполняется следующая последовательность операций:

1. Затем туннель можно активировать командой **ike**.
2. Демон **tmd** передает демону **isakmpd** запрос на установление соединения для управления ключами (первый этап).
3. Демон **isakmpd** возвращает состояние `SA создана` или сообщение об ошибке.
4. Демон **tmd** передает демону **isakmpd** запрос на установление соединения для управления данными (второй этап).
5. Демон **isakmpd** возвращает состояние `SA создана` или сообщение об ошибке.
6. Параметры туннеля добавляются в кэш туннелей ядра.

7. В динамическую таблицу фильтров ядра добавляются правила фильтрации.

Если система выполняет функции отвечающей стороны, то демон **isakmpd** уведомляет демона **tmd** администратора туннелей об успешном согласовании туннеля и добавляет в ядро новый туннель. В этом случае процесс начинается с шага 3 и продолжается до шага 7 без выдачи запросов на соединение демоном **tmd**.

Функция анализа полезной нагрузки:

Конфигурация защиты (SA) конечных точек туннеля устанавливается путем обмена сообщениями IKE. Функция анализа полезной нагрузки анализирует сообщения и представляет их в формате, удобном для человека.

Для включения ведения протокола функции анализа полезной нагрузки можно внести изменения в файл `/etc/isakmpd.conf`. Запись ведения протокола в файле `/etc/isakmpd.conf` выглядит примерно следующим образом:

```
information
```

Тип полезной нагрузки IKE для занесения в протокол этой функцией зависит от содержимого сообщения IKE. В качестве примеров можно назвать SA Payload, Key Exchange Payload, Certificate Request Payload, Certificate Payload и Signature Payload. Ниже приведен пример протокола анализа полезной нагрузки, в котором после ISAKMP_MSG_HEADER следует пять блоков полезной нагрузки:

```
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)
```

В каждом блоке полезной нагрузки (в поле **Next Payload**) указывается следующий блок. Если текущий блок последний в сообщении IKE, то в поле **Next Payload** указывается нулевое значение (None).

Каждый блок полезной нагрузки в данном примере содержит информацию, относящуюся к выполняемому согласованию. Например, блок SA Payload включает Proposal Payload и Transform Payload, в которых, в свою очередь, задаются предлагаемые отправителем алгоритм шифрования, режим идентификации, алгоритм хеширования, тип и продолжительность существования SA.

Кроме того, в SA Payload указывается один или несколько блоков Proposal Payload и один или несколько блоков Transform Payload. В поле **Next Payload** для Proposal Payload указывается значение 0, если это единственный блок Proposal Payload, либо значение 2, если после этого блока следует еще один или несколько блоков Proposal Payload. Аналогично, в поле **Next Payload** для Transform Payload указывается значение 0, если это единственный блок Transform Payload, либо значение 3, если после этого блока следует еще один или несколько блоков Transform Payload, как в следующем примере:

```
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)
```

В заголовке сообщения IKE блок Parse Payload указывает тип обмена (основной или ускоренный режим), полную длину сообщения, идентификатор сообщения и т.д.

Блок Certificate Request Payload запрашивает у отвечающей системы сертификат. Отвечающая система возвращает сертификат в отдельном сообщении. В следующем примере показаны блоки Certificate Payload и Signature Payload, передаваемы в процессе согласования SA. Данные сертификата и подписи указаны в шестнадцатеричном формате.

```
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
```

Неполадки режима подписи и цифрового сертификата:

В этом разделе перечисляются возможные неполадки режима подписи и цифровых сертификатов, а также варианты их устранения:

Ошибка	**Возможные неполадки и их решения**
Ошибка: не запускается **cpsd** (демон сервера Proxy сертификатов). В файл протокола заносится сообщение, аналогичное следующему: `Sep 21 6:02:00 ripple CPS[19950]: Init():Lo adCaCerts() failed, rc =-12`	Неполадка: База данных сертификатов не открыта или не найдена. Исправление: Убедитесь, что в `/etc/security` существует база данных сертификатов Диспетчера ключей. В состав этой базы данных входят следующие файлы: `ikekey.crl`, `ikekey.kdb`, `ikekey.rdb` и `ikekey.sth`. Если нет только файла `ikekey.sth`, значит, при создании базы данных не была выбрана опция `Сохранить пароль`. Для применения цифровых сертификатов с защитой IP пароль должен быть сохранен. (Дополнительная информация приведена в разделе Создание базы данных ключей.)

Ошибка: При получении сертификата администратор ключей выдает следующее сообщение:
 	
`Обнаружены недопустимые данные в формате Base64`
	Неполадка: В файле сертификата обнаружены избыточные данные, либо данные сертификата повреждены или утеряны.
	
	Исправление: Сертификат в формате 'DER' должен быть заключен между следующими строками. Никакие другие символы, кроме строк BEGIN CERTIFICATE и END CERTIFICATE, встречаться не должны.
	
	`-----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-----`
	
	Локализовать и устранить эту неполадку можно следующим образом.
	• Если данные утеряны или повреждены, то создайте сертификат заново.
	• С помощью анализатора ASN.1, который можно загрузить из Internet, проверьте правильность сертификата.

Ошибка: при получении личного сертификата администратор ключей выдает следующее сообщение: `Не найден ключ запроса сертификата.`	Неполадка: Для полученного личного сертификата не найден запрос. Исправление: Создайте новый запрос на получение личного сертификата и запросите новый сертификат.
Ошибка: согласование IKE не выполняется и в файл протокола заносится сообщение, аналогичное следующему: `inet_cert_service:: channelOpen(): clientInitIPC():error,rc =2 (No such file or directory)`	Неполадка: Демон **cpsd** не запущен или остановлен. Исправление: запустите защиту IP, которая запустит соответствующие демоны.

Ошибка	Возможные неполадки и их решения
Ошибка: согласование IKE не выполняется и в файл протокола заносится сообщение, аналогичное следующему: `CertRepo::GetCertObj: Недопустимое DN: ("/C=US/O=IBM/ CN=ripple.austin.ibm.com")`	Неполадка: При определении туннеля IKE указано отличительное имя (DN) X.500, не соответствующее DN X.500, заданному в личном сертификате. Исправление: Измените определение туннеля IKE таким образом чтобы отличительное имя соответствовало указанному в сертификате.

Утилиты трассировки:

Трассировка событий ядра предназначена для отладки. Трассировка позволяет получить множество информации о различных событиях и ошибках, связанных с кодом поддержки туннелей и фильтров.

Утилите трассировки защиты IP можно обратиться с помощью меню SMIT Расширенная настройка защиты IP. Эта утилита позволяет собирать следующую информацию: ошибки, фильтры, информация о фильтрах, туннели, информация о туннелях, капсуляция/декапсуляция, информация о капсуляции, шифрование и информация о шифровании. По умолчанию наиболее самая важная информация собирается при выборе опции Ошибка. При выборе опции Информация возможен сбор критически важной информации, однако при этом может снизиться производительность системы. Эта трассировка предоставляет информацию о неполадке, а также требуется для объяснения неполадки технику по обслуживанию.

Для включения трассировки настройте устройства IPSec и задайте уровень трассировки каждого подкомпонента IPSec на уровень трассировки 7 для получения достаточных данных трассировки ядра. Если устройства IPSec не настроены, то команда управления трассировкой компонентов не включает записи, связанные с IPSec. Для запуска трассировки IPSec используйте команду быстрого доступа SMIT **smit ips4_start** (для IP версии 4) или **smit ips6_start** (для IP версии 6).

Примечание: Если трассировка компонентов IPSec не задана правильно, то записи трассировки будут пустыми.

Для получения данных трассировки ядра выполните следующее:

1. Запросите все компоненты для просмотра текущих параметров уровней трассировки:
   ```
   # ctctrl -q
   ```
2. Проверьте компонент IPSec и подкомпоненты. Первоначально компоненты с уровнем трассировки 3 по умолчанию выглядят следующим образом. Для просмотра первоначальной уровня трассировки по умолчанию введите:
   ```
   # ctctrl -q -c ipsec -r
   ```

Имя компонента	Наличие псевдонима	Трассировка памяти/уровень	Трассировка системы/уровень	Размер буфера/Выделенный
ipsec	NO	ВКЛ./3	ВКЛ./3	4Д0960/Да
.capsulate	НЕТ	ВКЛ./3	ВКЛ./3	10240/ДА
.filter	НЕТ	ВКЛ./3	ВКЛ./3	10240/ДА
.tunnel	НЕТ	ВКЛ./3	ВКЛ./3	10240/ДА

3. Увеличьте уровень IPSec и подкомпонентов до 7 для поддержки трассировки ядра, введите:
   ```
   # ctctrl systracelevel=7 -c ipsec -r
   ```
4. Для запроса подтверждения изменения уровней трассировки для IPSec и подкомпонентов введите:
   ```
   # ctctrl -q -c ipsec -r
   ```

Имя компонента	Наличие псевдонима	Трассировка памяти/уровень	Трассировка системы/уровень	Размер буфера/Выделенный
ipsec	НЕТ	ВКЛ./3	ВКЛ./7	4Д0960/Да
.capsulate	НЕТ	ВКЛ./3	ВКЛ./7	10240/ДА
.filter	НЕТ	ВКЛ./3	ВКЛ./7	10240/ДА
.tunnel	НЕТ	ВКЛ./3	ВКЛ./7	10240/ДА

Для вызова утилиты трассировки воспользуйтесь командой SMIT **smit ips4_tracing** (для IPv4) или **smit ips6_tracing** (для IPv6). Трассировки ядра через **smit ips4_tracing**, **smit ips6_tracing** средство трассировки командной строки дают достаточные данные трассировки IPSec.

Команда ipsecstat:

С помощью команды **ipsecstat** можно посмотреть список устройств, алгоритмы шифрования и статистику пакетов IP Security packets.

При работе команды **ipsecstat** создается отчет, в котором показано, что устройства защиты IP доступны, установлено три алгоритма идентификации, три алгоритма шифрования и что есть текущий отчет об операциях с пакетами. Эта информация может пригодиться для определения и устранения неполадок защиты IP.

```
Устройства защиты IP:
ipsec_v4 доступно
ipsec_v6 доступно

Алгоритмы идентификации:
HMAC_MD5 -- Hashed MAC MD5 Authentication Module
HMAC_SHA -- Hashed MAC SHA Hash Authentication Module
KEYED_MD5 -- Keyed MD5 Hash Authentication Module

Алгоритмы шифрования:
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

Статистика защиты IP -
Всего получено пакетов: 1106
Получено пакетов AH: 326
Получено пакетов ESP: 326
Разрешено пакетов Srcrte: 0
Всего отправлено пакетов: 844
Отправлено пакетов AH: 527
Отправлено пакетов ESP: 527
Всего отброшено полученных пакетов: 12
  Отброшено фильтром на входе: 12
  Неполные AH: 0
  Неполные ESP: 0
  Нарушение передачи AH: 0
  Нарушение передачи ESP: 0
Всего отброшено исходящих пакетов: 0
  Отброшено фильтром на входе: 0
Добавлено записей в кэш туннелей: 7
Истечение срока хранения записей кэша туннелей: 0
Удалено записей из кэша туннелей: 6
```

Примечание: В применении CDMF нет необходимости, поскольку сейчас широкое распространение получил DES. Измените конфигурацию туннелей, использующих CDMF, перейдя к применению DES или Triple DES.

Справочник по защите IP

Существуют команды и методы для защиты IP. Также можно осуществить перенос туннелей, фильтров и общих ключей IKE.

Перечень команд:

В этой таблице приведен перечень команд.

Команда	Назначение
ike cmd=activate	Запускает согласование IKE.
ike cmd=remove	Деактивирует туннели IKE
ike cmd=list	Выводит список туннелей IKE
ikedb	Предоставляет интерфейс для работы с базой данных туннелей IKE
gentun	Создает определение туннеля
mktun	Активирует определения туннелей
chtun	Изменяет определение туннеля
rmtun	Удаляет определение туннеля
lstun	Показывает список определений туннелей
exptun	Экспортирует определения туннелей
imptun	Импортирует определения туннелей
genfilt	Создает определение фильтра
mkfilt	Активирует определения фильтров
mvfilt	Перемещает правило фильтрации
chfilt	Изменяет определение фильтра
rmfilt	Удаляет определение фильтра
lsfilt	Показывает список определений фильтров
expfilt	Экспортирует определения фильтров
impfilt	Импортирует определения фильтров
ipsec_convert	Показывает информацию о состоянии защиты IP
ipsecstat	Показывает информацию о состоянии защиты IP
ipsectrcbuf	Показывает содержимое буфера трассировки защиты IP
unloadipsec	Выгружает модуль шифрования

Список методов:

Ниже приведен список методов.

defipsec
Определяет экземпляр защиты IP для IP версии 4 или версии 6

cfgipsec
Настраивает и загружает **ipsec_v4** или **ipsec_v6**

ucfgipsec
Удаляет конфигурацию **ipsec_v4** или **ipsec_v6**

Перенос конфигурации защиты IP:

Туннели IKE, фильтры и подготовленные ключи можно переносить из более ранних версий операционной системы AIX.

Перенос туннелей IKE:

Для переноса туннелей выполните следующие действия:

1. Запустите сценарий `bos.net.ipsec.keymgt.pre_rm.sh`. При запуске этого сценария в каталоге `/tmp` создаются следующие файлы:
   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`

Внимание: Запустите этот сценарий только один раз. В случае повторного запуска сценария после обновления базы данных все файлы будут утеряны без возможности их восстановления. Перед переносом туннелей в новую версию ознакомьтесь со сценарием, описанным в разделе “Сценарий bos.net.ipsec.keymgt.pre_rm.sh” на стр. 276.

2. Сохраните файлы, созданные сценарием, и файл `/tmp/lpplevel` на каком-либо внешнем носителе, например, на компакт-диске или дискете.

Перенос подготовленных ключей:

В этом разделе приведена инструкция по обновлению формата подготовленных ключей.

База данных подготовленных ключей туннелей IKE повреждается в процессе перехода к новой версии. Для обновления формата подготовленного ключа выполните следующие действия в перенесенной системе:

1. Сохраните вывод команды **ikedb -g**, запустив следующую команду:
   ```
   ikedb -g > out.keys
   ```
2. Откройте файл `out.keys` и для формата подготовленного ключа замените FORMAT=ASCII на FORMAT=HEX.
3. Загрузите файл XML с помощью следующей команды:
   ```
   ikedb -pF out.keys
   ```

Перенос фильтров:

Для переноса фильтров выполните следующие действия.

1. Экспортируйте файлы с правилами фильтрации в каталог `/tmp` с помощью SMIT, выполнив следующие действия:
   a. Введите команду **smitty ipsec4**.
   b. Выберите `Расширенная настройка защиты IP->Настроить правила фильтрации защиты IP->Экспортировать правила фильтрации защиты IP`.
   c. В качестве имени каталога введите `/tmp`.
   d. В поле Правила фильтрации нажмите F4 и выберите из списка **все**.
   e. Нажмите Enter, чтобы сохранить правила фильтрации в файле `/tmp/ipsec_fltr_rule.exp` на внешнем носителе.

   Выполните этот процесс для всех систем, для которых выполняется миграция с прежних версий операционной системы AIX.
2. Скопируйте шесть файлов туннелей, созданных сценарием, а также файлы `/tmp/lpplevel` и `/tmp/ipsec_fltr_rule.exp` в каталог `/tmp` обновленной системы.
3. Запустите сценарий `bos.net.ipsec.keymgt.post_i.sh` для загрузки конфигураций туннелей в базу данных.
4. Введите команду **ikedb -g**, чтобы проверить наличие туннелей в базе данных.

   Примечание: Если информация о туннелях не видна в базе данных, запустите сценарий повторно, предварительно заменив имена всех файлов `*.loaded` в каталоге `/tmp` их первоначальными именами.

После перехода база данных фильтров будет повреждена. При запуске команды **lsfilt** в обновленной системе будет выдаваться следующее сообщение об ошибке:

```
Невозможно получить правило фильтрации ipv4 по умолчанию
```

Для обновления базы данных фильтров выполните следующие действия:

1. Замените файлы `ipsec_filter` и `ipsec_filter.vc` в каталоге `/etc/security` неповрежденными файлами. Если таких файлов нет, получите их в сервисном представительстве фирмы IBM.

2. Импортируйте файлы с правилами фильтрации в каталог /tmp с помощью SMIT, выполнив следующие действия:
   a. Введите команду **smitty ipsec4**.
   b. Выберите Расширенная настройка защиты IP->Настроить правила фильтрации защиты IP->Импортировать правила фильтрации защиты IP.
   c. В качестве имени каталога введите /tmp.
   d. В поле **Правила фильтрации** нажмите **F4** и выберите из списка **все**.
   e. Для повторного создания правил фильтрации нажмите Enter. Список правил фильтрации можно просмотреть с помощью SMIT или команды **lsfilt**.

Сценарий bos.net.ipsec.keymgt.pre_rm.sh:

Сценарий bos.net.ipsec.keymgt.pre_rm.sh сохраняет содержимое базы данных туннеля в системе, работающей под управлением 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


#  Скопировать базу данных в каталог сохранения на случай ошибки при изменении
if [ -d /etc/ipsec/inet/DB ]
then
  cp -R /etc/ipsec/inet/DB /etc/ipsec/inet/DB.sav || exit $?
fi

#  Запомнить уровень, с которого переносятся данные
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

#  Выяснить, существует ли ikedb.
if [ -f $IKEDB ]
then

  # Если какой-либо из вызовов ikedb окажется неуспешным, ничего страшного.  Удалите
  # полученный файл (он может содержать ненужные данные) и продолжите операцию.  Сценарий
  # post_i просто не импортирует несуществующий файл, в результате чего
  # база данных IKE или ее часть будет утеряна, но это более предпочтительный вариант,
  # чем выход из сценария с кодом ошибки и сбою всей процедуры
  # миграции.

  $IKEDB -g > $XMLFILE
  if [ $? -ne 0 ]
  then
    rm -f $XMLFILE  || exit $?
  fi

  if [[ $VR = "5.1" ]]; then
    # Это специальный случай.  5.1 - это единственная версия ikedb,
    # в которой подготовленные ключи не включаются в вывод всей
    # базы данных.  Поэтому их необходимо извлекать по очереди.
    $IKEDB -g -t IKEPresharedKey > $PSKXMLFILE
    if [ $? -ne 0 ]
```

```
    then
     rm -f $PSKXMLFILE  || exit $?
    fi
  fi

#  Проверка наличия команды ikegui
elif [ -f /usr/sbin/ikegui ]
then

  #  Получение информации о базе данных и ее сохранение в каталоге /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
```

Сценарий bos.net.ipsec.keymgt.post_i.sh:

Сценарий bos.net.ipsec.keymgt.post_i.sh загружает содержимое базы данных туннелей в обновленную систему с операционной системой 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
```

Защита сетевой файловой системы

Сетевая файловая система (NFS) - это широко распространенная технология, позволяющая различным хостам сети работать с общими данными.

В NFS, наряду с DES, также поддерживается использование идентификации Kerberos 5. Для обеспечения защиты Kerberos 5 применяется протокол RPCSEC_GSS.

Помимо стандартной системы идентификации UNIX в NFS предусмотрены собственные средства сетевой идентификации пользователей и систем на уровне сообщения. Эти средства идентификации используют стандарт шифрования DES с общим ключом.

В NFS, наряду с DES, также поддерживается использование идентификации Kerberos 5. Для обеспечения защиты Kerberos 5 применяется протокол RPCSEC_GSS. Сведения об администрировании и применении идентификации Kerberos в NFS приведены в книге *Руководство по администрированию NFS*.

Общее руководство по защите сетевой файловой системы

В этом разделе приведено руководство по защите сетевой файловой системы (NFS).

- Убедитесь, что установлены последние версии исправлений программного обеспечения. Особое внимание обратите на исправления, связанные с защитой системы. Следует выполнять обслуживание всего программного обеспечения, образующего единую инфраструктуру. Например, установив исправления для операционной системы, но оставив Web-сервер без исправлений, можно предоставить злоумышленнику возможность для проникновения в систему, чего можно было бы избежать, своевременно обновив Web-сервер. Для того чтобы подписаться на предупреждения о защите IBM System p для получения новейшей доступной информации о безопасности посетите следующий веб-адрес: http://www14.software.ibm.com/webapp/set2/subscriptions/pqvcmjd.
- Настройте сервер NFS таким образом, чтобы файловые системы экспортировались с минимальным набором прав доступа. Если пользователям необходимо только считывать информацию из файловой системы, у них не должно быть прав на запись в эту файловую систему. Это позволит предотвратить

попытки заменить важные данные, изменить файлы конфигурации или записать вредоносный исполняемый код в экспортируемую файловую систему. Указывайте права доступа с помощью SMIT или непосредственно в файле `/etc/exports`.

- Экспортируйте файловые системы на сервере NFS только для тех пользователей, которым необходим доступ к этим файловым системам. В большинстве реализаций NFS можно указать, каким клиентам NFS разрешен доступ к той или иной файловой системе. Это позволит сократить количество попыток несанкционированного доступа к файловым системам. В частности, не экспортируйте файловую систему на тот сервер NFS, с которого она экспортируется.
- Экспортируемые файловые системы должны располагаться в отдельных разделах. Злоумышленник может добиться снижения производительности системы, записывая данные в экспортированную файловую систему до тех пор, пока она не переполнится. В результате файловая система станет недоступна для тех приложений и пользователей, которым она действительно необходима.
- Запретите клиентам NFS обращаться к файловой системе от имени пользователя root или от имени неизвестного пользователя. В большинстве реализаций NFS можно настроить преобразование запросов привилегированных и неизвестных пользователей в запросы обычных пользователей. Это не даст возможность злоумышленнику получать доступ к файлам и выполнять операции над файлами от имени привилегированного пользователя.
- Запретите клиентам NFS запускать программы suid и sgid в экспортированных файловых системах. Это позволит предотвратить выполнение вредоносного кода с правами администратора. Если злоумышленнику удастся создать исполняемый файл, владельцем которого является привилегированный пользователь или группа, серверу NFS может быть нанесен значительный ущерб. Для того чтобы установить этот запрет, укажите опцию команды **mknfsmnt -y**.
- Используйте защищенную NFS. В защищенной NFS для идентификации хостов, участвующих в транзакциях RPC, применяется шифрование DES. RPC - это протокол, используемый NFS для обмена запросами между хостами. Защищенная NFS зашифровывает системное время в запросах RPC и тем самым снижает вероятность того, что злоумышленнику удастся имитировать запрос RPC. Расшифровав системное время и проверив его правильность, получатель может убедиться в том, что запрос RPC получен от надежного хоста.
- Если вы не планируете применять NFS, выключите ее. Это снизит количество возможных путей для атаки.

В NFS в дополнение к Triple DES и Single DES поддерживается шифрование AES с идентификацией Kerberos 5. Сведения о настройке Kerberos 5 для использования типа шифрования AES приведены в Руководстве по администрированию NFS.

Понятия, связанные с данным:

“Защита сетевой файловой системы” на стр. 281

Информация, связанная с данной:

Справочная таблица по настройке NFS

Запуск демонов NFS при запуске системы

Настройка сервера NFS

Настройка клиента NFS

Преобразование идентификаторов

Экспорт файловой системы NFS

Настройка сети для RPCSEC-GSS

Отмена экспорта файловой системы NFS

Изменение экспортированной файловой системы

Доступ пользователя root к экспортированной файловой системе

Монтирование файловой системы NFS вручную

Подсистема Automount

Установка предопределенных монтирований NFS

Удаление предопределенных монтирований NFS

выполняет экспорт файла для NFS

команда mknfsmnt

Идентификация в сетевой файловой системе

В NFS алгоритм шифрования DES применяется в различных целях. Алгоритм DES применяется для шифрования меток времени в сообщениях вызова удаленных процедур (RPC), которыми обмениваются серверы и клиенты NFS. Зашифрованные метки времени применяются в качестве маркеров из приведенного выше примера.

Отдельная идентификация каждого сообщения RPC обеспечивает еще более высокий уровень защиты каждой файловой системы. По умолчанию файловые системы экспортируются в стандартном режиме идентификации UNIX. Для того чтобы воспользоваться усовершенствованными средствами идентификации, нужно указать при экспорте файловой системы опцию secure.

Шифрование с общим ключом в защищенной NFS:

Общие и личные ключи хранятся в базе данных `publickey.byname`, проиндексированной по имени сети.

Личные ключи зашифрованы по алгоритму DES с паролем пользователя в качестве ключа. Команда **keylogin** расшифровывает личный ключ с помощью пароля и передает его на защищенный локальный сервер ключей для применения в транзакциях RPC. Общие и личные ключи неизвестны пользователям, поскольку они создаются командой **yppasswd** автоматически.

Демон keyserv представляет собой службу RPC, работающую во всех системах NIS. В NIS, команда **keyserv** выполняет следующие относящиеся к шифрованию с общим ключом процедуры:

- **key_setsecret**
- **key_encryptsession**
- **key_decryptsession**

Процедура **key_setsecret** сохраняет личный ключ пользователя (SK_A) на сервере ключей. Обычно эта процедура вызывается командой **keylogin**. Программы клиента пользуются процедурой **key_encryptsession** для создания зашифрованного ключа, который будет применяться в ходе сеанса связи и передается на сервер при выполнении первой транзакции RPC. Сервер ключей объединяет общий ключ сервера и личный ключ клиента (созданный ранее с помощью процедуры **key_setsecret**) и создает общий ключ. Затем сервер дешифрует ключ сеанса с помощью процедуры **key_decryptsession**.

В этих процедурах неявно используется имя вызывающего субъекта, которое также должно быть идентифицировано. На этом этапе нельзя использовать алгоритм DES, так как это привело бы к тупиковой ситуации. Поэтому личные ключи пользователей хранятся в базе данных, а запросы на ключи принимаются только от локальных процессов пользователя root. Затем процесс клиента вызывает от имени root процедуру **setuid** и сообщает серверу ключей реальный идентификатор клиента.

Требования к идентификации в сетевой файловой системе:

Система идентификации защищенной NFS основана на том, что отправитель шифрует текущее время, а получатель дешифрует это значение и сравнивает с показаниями собственных часов.

Для выполнения этой процедуры должны быть выполнены следующие требования:

- У отправителя и получателя должны быть синхронизированы показания часов.
- Отправитель и получатель должны пользоваться одним и тем же ключом шифрования DES.

Синхронизация часов:

Если в сети есть внутренние средства синхронизации часов, то показания часов сверяются с помощью демона timed. В противном случае клиент синхронизирует свои часы с сервером.

Для этого клиент перед запуском сеанса RPC клиент считывает показания часов сервера и сравнивает разницу в показаниях своих часов и часов сервера. После этого клиент синхронизирует свои часы с сервером. Если в ходе сеанса RPC синхронизация будет нарушена и сервер начнет отклонять запросы клиента, клиент проведет повторную синхронизацию.

Работа с единым ключом DES:

Клиент и сервер пользуются единым ключом DES и передают этот ключ друг другу средствами шифрования с открытым ключом.

Для каждого клиента A и сервера B существует ключ, называемый *единым ключом*, который известен только A и B. Клиент получает единый ключ по следующей формуле:

$$K_{AB} = PK_B^{SK}A$$

где *K* - это единый ключ, *PK* - общий ключ, а*SK* - закрытый ключ. Длина всех ключей - 128 разрядов. Сервер получает тот же самый единый ключ по следующей формуле:

$$K_{AB} = PK_A^{SK}B$$

Поскольку для определения единого ключа требуются личные ключи сервера и клиента, этот ключ известен только серверу и клиенту. Поскольку длина единого ключа составляет 128 разрядов, а в алгоритме DES применяются 56-разрядные ключи, клиент и сервер формируют ключ DES путем извлечения 56 разрядов из единого ключа.

Процесс идентификации в сетевой файловой системе:

Когда клиенту нужно установить связь с сервером, он выбирает произвольный ключ для шифрования меток времени. Этот ключ называется *ключом сеанса* (*CK*).

Клиент шифрует ключ сеанса с помощью единого ключа DES (см. раздел Требования к системе идентификации) и передает его серверу в рамках первой транзакции RPC. Эта процедура проиллюстрирована на следующем рисунке.

Рисунок 15. Процедура идентификации. На этом рисунке проиллюстрирована процедура идентификации, описанная в этом разделе.

На рисунке показано соединение клиента A и сервера B. Термин *K*(*CK*) означает, что ключ *CK* зашифрован с помощью единого ключа DES *K*. В первом запросе в ходе сеанса в идентификационных данных клиента для RPC передается имя клиента (*A*), ключ сеанса (*CK*) и переменная *win* (окно), зашифрованная с помощью ключа *CK*. (По умолчанию размер окна составляет 30 минут.) Идентификатор клиента в первом запросе содержит зашифрованную метку времени и зашифрованный идентификатор указанного окна (*win* + 1). Наличие идентификатора окна значительно затрудняет подбор идентификационной информации и повышает надежность защиты.

После проверки клиента сервер заносит следующие данные в таблицу разрешений:

- Имя клиента *A*
- Ключ сеанса *CK*
- Окно
- Метка времени

Сервер принимает метки времени только в возрастающем порядке, и это обеспечивает невозможность повтора уже выполненных транзакций. Сервер возвращает клиенту идентификатор с номером позиции в таблице разрешений и меткой времени клиента, уменьшенной на 1 и зашифрованной с помощью *CK*. Клиенту известно, что такой идентификатор мог быть отправлен только сервером, поскольку только серверу известна метка времени, отправленная клиентом. Вычитание единицы из метки времени выполняется для того, чтобы данная метка времени стала устаревшей и не могла использоваться в качестве идентификатора клиента. После выполнения первой транзакции RPC клиент передает серверу только свой идентификатор в таблице разрешений и зашифрованную метку времени, а сервер возвращает клиенту только метку времени, уменьшенную на 1 и зашифрованную с помощью ключа *CK*.

Имена объектов сети в системе идентификации DES

В системе идентификации DES всем объектам присвоены сетевые имена.

Сетевое имя - это последовательность символов, используемая в качестве идентификатора. Общие и личные ключи привязываются не к пользователям, а к сетевым именам. Информация о соответствии сетевых имен и локальных идентификаторов пользователей и групп хранится в базе данных `netid.byname` NIS.

Имена пользователей уникальны в пределах домена. Сетевые имена формируются путем слияния идентификаторов операционной системы и пользователя с доменными именами NIS и Internet. Рекомендуется добавлять к имени локального домена имя домена Internet (например, com, edu, gov, mil).

Сетевые имена присваиваются не только пользователям, но и системам. Сетевое имя системы формируется по подобию сетевого имени пользователя. Например, системе `hal` в домене `eng.xyz.com` присваивается

сетевое имя unix.hal@eng.xyz.com. Возможность идентификации систем особенно важна для бездисковых клиентов, которым нужен доступ к своим домашним каталогам, расположенным в сети.

Для идентификации пользователей удаленных доменов сведения о них записываются в две базы данных NIS. В первой базе данных хранятся общие и личные ключи, а во второй - локальные идентификаторы пользователей и списки управления доступом. Таким образом можно предоставить пользователям из других доменов любой доступ к локальным сетевым службам, например, NFS или rlogin.

Файл /etc/publickey

В файле /etc/publickey хранятся имена и общие ключи, применяемые NIS для создания базы данных publickey.

База данных publickey служит для организации защищенного сетевого пространства. В каждой записи этого файла указаны сетевое имя пользователя или хоста, общий ключ (в шестнадцатеричном формате), двоеточие и зашифрованный личный ключ (также в шестнадцатеричном формате). По умолчанию в файле /etc/publickey есть только пользователь nobody.

Не следует изменять файл /etc/publickey с помощью обычных текстовых редакторов, так как в нем содержится зашифрованная информация. Изменять файл /etc/publickey следует с помощью команд **chkey** или **newkey**.

Замечания о загрузке системы с общими ключами

При перезагрузке системы из-за сбоя питания все личные ключи теряются, и защищенные сетевые службы становятся недоступны клиентам. Процессы root могут продолжить работу, если кто-нибудь сможет указать пароль для дешифрования личного ключа пользователя root. Поэтому рекомендуется хранить расшифрованный личный ключ пользователя root в каком-либо файле, доступном серверу ключей.

Не все вызовы процедуры **setuid** работают правильно. Например, если функция **setuid** будет вызвана пользователем-владельцем A, который еще не входил в систему с момента ее запуска, этой функции будут недоступны защищенные сетевые службы от имени пользователя A. Однако чаще всего функция **setuid** вызывается пользователем root, личный ключ которого всегда сохраняется в момент запуска системы.

Влияние защищенной NFS на производительность системы

Защищенная NFS может оказывать влияние на производительность системы различными способами.

- Сначала клиенту и серверу требуется вычислить единый ключ. На это требуется около одной секунды. Поэтому на установку первого соединения RPC требуется около двух секунд. После этого сервер ключей заносит результаты вычислений в кэш, и в дальнейшем вычислять единый ключ не требуется.
- Для выполнения каждой транзакции RPC требуется выполнить следующие операции:
  1. Клиент шифрует метку времени для запроса.
  2. Сервер расшифровывает это значение.
  3. Сервер шифрует ответную метку времени.
  4. Клиент расшифровывает это значение.

Поскольку защищенная NFS снижает производительность системы, вам следует сопоставить преимущества повышения степени защиты и степень снижения производительности.

Справочная таблица защищенной сетевой файловой системы

Следующая таблица предназначена для проверки конфигурации защищенной NFS.

- При монтировании файловой системы с опцией **-secure** имя сервера должно соответствовать имени хоста сервера, указанному в файле /etc/hosts. Если в вашей сети применяется сервер DNS, убедитесь, что возвращаемое им значение совпадает с записью в файле /etc/hosts. Если эти имена не будут совпадать, то неизбежны проблемы идентификации, так как сетевые имена назначаются системам по содержимому файла /etc/hosts, а выборка ключей из базы данных **publickey** осуществляется по сетевым именам.

- Не следует одновременно пользоваться защищенным и незащищенным импортом и экспортом. Это может привести к неправильному определению прав доступа к файлам. Например, если клиент попытается смонтировать защищенную файловую систему без опции **-secure** или незащищенную файловую систему с опцией **-secure**, то ему будут предоставлены права доступа `nobody`, а не его собственные права доступа. Эта же ситуация возникнет в случае, если пользователь, неизвестный в NIS, попытается создать или изменить файлы в защищенной файловой системе.
- Так как служба NIS должна распространять новую версию базы данных после каждого выполнения команд **chkey** и **newkey**, не рекомендуется работать с этими командами в периоды интенсивной загрузки сети.
- Не удаляйте файлы `/etc/keystore` и `/etc/.rootkey`. Если вы планируете перемещать или модернизировать систему, или переустанавливать операционную систему на ней, сохраните файлы `/etc/keystore` и `/etc/.rootkey`.
- Проинструктируйте пользователей о том, что вместо команды **ppasswd** для изменения пароля следует применять команду **yppasswd**. В противном случае может нарушиться синхронизация паролей и личных ключей.
- Так как команда **login** не получает ключи демона **keyserv** из базы данных publickey, вместо нее следует применять команду **keylogin**. Команду **keylogin** можно занести в файлы `profile` всех пользователей, чтобы она автоматически выполнялась при входе в систему. При выполнении команды **keylogin** пароль запрашивается повторно.
- После создания ключей для пользователя root с помощью команд **newkey -h** и **chkey** нужно передать новые ключи демону **keyserv** с помощью команды **keylogin**. Ключи хранятся в файле `/etc/.rootkey`, который читается демоном **keyserv** при каждом его запуске.
- Регулярно проверяйте, работают ли демоны **yppasswdd** и **ypupdated** на главном сервере NIS. Эти демоны обслуживают базу данных publickey.
- Регулярно проверяйте, работает ли демон **keyserv**, на всех системах, использующих защищенную NFS.

Настройка защищенной сетевой файловой системы

Для настройки защищенной NFS на главном и подчиненных серверах NIS используйте описанный ниже алгоритм.

1. На главном сервере NIS нужно создать записи для всех пользователей в файлеNIS `/etc/publickey` с помощью команды**newkey**:
   - Для обычных пользователей выполните следующую команду:

     ```
     smit newkey
     ```

     или

     ```
     newkey -u имя-пользователя
     ```

     Для пользователя root выполните следующую команду:

     ```
     newkey -h имя-хоста
     ```
   - Кроме того, с помощью команд **chkey** или **newkey** пользователи могут создавать свои ключи.
2. Создайте базу данных NIS publickey. Соответствующая база данных `publickey.byname` NIS хранится только на серверах NIS.
3. Удалите символы комментария для следующих разделов файла `/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. Запустите демоны **keyserv**, **ypupdated** и **yppasswdd** командой **startsrc**.

Для того чтобы настроить защищенную NFS на клиентах NIS запустите демон **keyserv** с помощью команды **startsrc**.

Экспорт файловой системы с помощью защищенной NFS

Защищенную NFS можно экспортировать с помощью одной из следующих процедур.

- Для того чтобы экспортировать файловую систему защищенной NFS с помощью SMIT, выполните следующие действия:
  1. С помощью команды **lssrc -g nfs** проверьте, работает ли служба NFS. В выводе команды демоны nfsd и rpc.mountd должны отображаться как активные.
  2. Проверьте, существует ли база данных publickey, и запущен ли демон keyserv. Дополнительные сведения приведены в разделе “Настройка защищенной сетевой файловой системы” на стр. 287.
  3. Введите команду **smit mknfsexp**.
  4. Укажите путь к экспортируемому каталогу, режим экспорта каталога и время экспорта (сейчас, при загрузке системы или и то, и другое). Укажите в поле Защищенная значение Да.
  5. Укажите прочие дополнительные параметры или оставьте значения по умолчанию.
  6. Завершите работу SMIT. Если файл `/etc/exports` не существовал до этого, он будет создан.
  7. Повторите шаги 3-6 для всех каталогов, которые требуется экспортировать.
- Для того чтобы экспортировать файловую систему защищенной NFS с помощью текстового редактора, выполните следующие действия:
  1. Откройте файл `/etc/exports` в текстовом редакторе.
  2. Добавьте в этот файл записи для всех каталогов, которые нужно экспортировать. Не забудьте, что нужно указывать полные пути к каталогам. Каждый каталог нужно указывать в новой строке. Ни один экспортируемый каталог не должен быть вложенным в другой экспортированный каталог. Формат файла `/etc/exports`, включая описание опции `secure`, описан в документации по файлу /etc/exports.
  3. Сохраните файл `/etc/exports` и выйдите из текстового редактора.
  4. Если служба NFS уже запущена, выполните следующую команду:

     ```
     /usr/sbin/exportfs -a
     ```

     Опция **-a** указывает команде **exportfs**, что нужно передать ядру NFS все содержимое файла `/etc/exports`.
- Для того чтобы временно экспортировать файловую систему (не изменяя файл `/etc/exports`), выполните следующие действия:

  ```
  exportfs -i -o secure /путь
  ```

  где `путь` - путь к экспортируемой файловой системе. Команда **exportfs -i** не проверяет наличие каталогов в файле `/etc/exports` и выполняет операцию в соответствии с параметрами, указанными в командной строке.

Экспорт файловой системы с помощью защищенной NFS

Каталог защищенной NFS можно явно смонтировать.

Для того чтобы явно смонтировать каталог защищенной NFS, выполните следующие действия:

1. Проверьте, экспортирован ли каталог NFS, с помощью следующей команды:

   ```
   showmount -e сервер
   ```

   где `сервер` - имя сервера NFS. Эта команда показывает список каталогов, экспортированных на сервере NFS в данный момент. Если каталог, который требуется смонтировать, не показан в списке, экспортируйте его с сервера.
2. Создайте локальную точку монтирования с помощью команды **mkdir**. Для того чтобы смонтировать каталог NFS, нужен каталог, который будет выполнять функцию точки монтирования. Этот каталог должен быть пустым. Точка монтирования создается точно так же, как любой другой каталог, никакие особые атрибуты указывать не нужно.

3. Проверьте, существует ли база данных publickey, и запущен ли демон keyserv. Дополнительные сведения приведены в разделе “Настройка защищенной сетевой файловой системы” на стр. 287.
4. Введите

   ```
   mount -o secure сервер:/remote/directory /local/directory
   ```

   где `сервер` - имя сервера NFS, `/remote/directory` - каталог на сервере NFS, который требуется смонтировать, и `/local/directory` - точка монтирования на клиенте NFS.

   Примечание: Монтировать каталоги защищенной NFS разрешено только пользователю root.

Преобразование идентификаторов предприятия

Современные сети состоят из сложных групп систем и приложений, требующих поддержания нескольких реестров пользователей. Работа с несколькими реестрами пользователей быстро превращается в серьезную проблему, влияющую и на самих пользователей, и на администраторов, и на разработчиков приложений. Функция преобразования идентификаторов в рамках предприятия (EIM) позволяет администраторам и разработчикам приложений быстро решать эту проблему.

В этом разделе рассказывается о возможных проблемах, кратко описываются самые распространенные способы их решения, а также описывается решение с помощью EIM.

Управление несколькими реестрами пользователей

Многим администраторам приходится управлять сетями, включающими самые разные системы, каждая из которых применяет свой собственный реестр пользователей и собственные процедуры управления этим реестром.

В таких сложных сетях администраторам приходится обеспечивать обслуживание идентификационных данных пользователей сразу в нескольких системах. Кроме того, администраторам часто приходится обеспечивать синхронизацию имен, паролей и прав доступа пользователей. Пользователям также приходится держать в голове множество имен и паролей, а также заботиться об их синхронизации. Поскольку пользователям и администраторам в такой среде приходится затрачивать слишком много усилий, администраторы часто тратят свое высокооплачиваемое рабочее время не на управление предприятием, а на разрешение проблем, связанных с забытыми паролями неудачными попытками входа в систему.

Проблема обслуживания нескольких реестров пользователей мешает жить и разработчикам приложений, создающих многоуровневые или неоднородные приложения. Важные данные заказчиков хранятся в самых разных системах, каждая из которых поддерживает только свой собственный реестр пользователей. В результате разработчикам приходится создавать для своих приложений собственный реестр пользователей и разрабатывать связанную с ним семантику защиты. Такой подход решает проблему разработчиков, но нисколько не облегчает жизнь пользователей и администраторов.

Современные подходы к преобразованию субъектов предприятия

В настоящее время существует несколько походов к решению проблемы управления несколькими реестрами пользователей, но ни один из них не обеспечивает полного решения этой проблемы. Например, простой протокол доступа к каталогам (LDAP) обеспечивает решение, в котором реализуется распределенный реестр пользователей. Однако, для применения решений, основанных на LDAP, администраторам придется обеспечить управление еще одним реестром пользователей и набором семантики защиты, либо заменить существующие приложения, обеспечив применение нового способа идентификации.

При использовании подобных решений администраторы вынуждены обслуживать несколько механизмов защиты различных ресурсов, что существенно повышает нагрузку на администраторов и увеличивает вероятность возникновения ошибок, приводящих к возникновению брешей в защите. Когда защита одного ресурса обеспечивается несколькими механизмами, всегда существует очень значительная вероятность того, что администратор изменит права доступа с помощью одного механизма и забудет сделать это для одного

или нескольких других механизмов. Например, ситуация, когда доступ пользователя к ресурсу запрещен через один интерфейс, но разрешен через один или несколько других, является потенциальной угрозой защите.

Тем не менее, после выполнения всего объема работы администраторы обнаружат, что проблема решена не полностью. Как правило, предприятия инвестируют довольно большой объем денег в свои реестры пользователей и связанную с ними семантику защиты, что делает реализацию подобного подхода неэффективной. Создание еще одного реестра пользователей со связанной семантикой решает проблему поставщика приложений, но не проблему пользователей и администраторов.

Другой подход заключается в применении механизма однократного входа в систему. Существует несколько продуктов, позволяющих администраторам управлять файлами, содержащими все идентификационные данные и пароли пользователя. Однако и у такого подхода есть свои недостатки:

- Проблема решается только с точки зрения пользователей. Несмотря на то, что для входа в несколько систем пользователь должен указать только одно имя и пароль, идентификационные данные должны храниться и обслуживаться во всех системах.
- Возникает потенциальная угроза безопасности, поскольку пароли в применяемых файлах обычно хранятся в текстовом или в легко расшифровываемом формате. Пароли всегда должны храниться в зашифрованном виде и не должны быть доступны другим пользователям, включая администраторов.
- Не решается проблема независимых поставщиков приложений, создающих неоднородные многоуровневые приложения. Они по-прежнему должны поддерживать в своих приложениях независимые реестры пользователей.

Несмотря на перечисленные недостатки, некоторые предприятия применяют такой подход, поскольку он несколько упрощает проблемы, связанные с наличием нескольких реестров пользователей.

Применение преобразования субъектов предприятия

Архитектура EIM описывает соотношения между отдельными объектами предприятия (например, файловыми серверами или серверами печати) и идентификаторами, соответствующими им в сети предприятия. Кроме того, EIM содержит набор API, позволяющих приложениям запрашивать информацию о таких взаимосвязях.

Например, если вам известно имя пользователя в одном из реестров, то вы можете определить, какая запись другого реестра соответствует этому пользователю. Если пользователь успешно прошел идентификацию с помощью одного реестра и вы можете установить соответствие между записью этого реестра и записью другого реестра пользователей, то пользователю не придется еще раз вводить свои данные для повторной идентификации. Все, что вам нужно знать - это какая запись соответствует данному пользователю в другом реестре. Таким образом, EIM обеспечивает общую функцию преобразования идентификаторов для всей сети предприятия.

Возможность установления соответствия между записями различных реестров пользователей обеспечивает множество преимуществ. Прежде всего, обеспечивается высокая гибкость - приложения могут применять для идентификации один реестр, а для проверки прав доступа - другой. Например, администратор может применять запись SAP для доступа к ресурсам SAP.

Преобразование идентификаторов требует от администратора выполнения следующих задач:

1. Создать идентификаторы EIM, соответствующие сотрудникам и объектам предприятия.
2. Создать определения реестров EIM, описывающие существующие реестры пользователей предприятия.
3. Определить взаимосвязь между идентификаторами пользователей в этих реестрах и созданными идентификаторами EIM.

Изменять программы обслуживания существующих реестров не требуется. Также не требуется обеспечивать преобразование всех идентификаторов из реестров пользователей. EIM обеспечивает преобразование один-несколько (другими словами, одному пользователю в реестре может соответствовать несколько идентификаторов). EIM также обеспечивает преобразование много-один (т.е. несколько пользователей могут

применять общий идентификатор из реестра), хотя применять такую конфигурацию и не рекомендуется по соображениям защиты. Администратор может указать в EIM любой реестр пользователей любого типа.

EIM не требует копирования существующих данных в новое хранилище и обеспечения синхронизации обеих копий. Единственный набор новых данных, необходимых для EIM, - это информация о взаимосвязи. Администраторы хранят эти сведения в каталоге LDAP, который обеспечивает необходимую гибкость, позволяя управлять данными в одном расположении, и создавать копии этих данных в тех местах, где информация применяется.

Kerberos

Kerberos - это служба сетевой идентификации, обеспечивающая идентификацию пользователей в физически незащищенных сетях. Kerberos обеспечивает взаимную идентификацию, а также гарантирует целостность и конфиденциальность данных в условиях, допускающих перехват, анализ и модификацию сетевого трафика.

Субъект Kerberos представляет собой уникальный идентификатор, используемый службами идентификации Kerberos. Kerberos проверяет идентификаторы, не используя механизмы идентификации операционной системы, требующие физической защиты всех хостов сети или доверия к адресам хостов.

Для идентификации в Kerberos применяются одноразовые разрешения и паспорта. Существует два вида паспортов: *начальные паспорта* и *служебные паспорта*. Начальный паспорт применяется в первоначальном запросе на идентификацию. При входе в систему для первоначальной идентификации может применяться, например, пароль или маркер. После получения начального паспорта паспорта вы можете воспользоваться им для запроса служебных паспортов различных служб. В таком механизме идентификации с помощью двух паспортов применяется *доверенная третья сторона* Kerberos. Паспорт на выдачу паспорта идентифицирует вас при обращении к серверу Kerberos, а служебный паспорт обеспечивает защищенную передачу информации о вас серверу.

Доверенная третья сторона или посредник Kerberos называется *центром рассылки ключей* (KDC). Он выдает клиентам паспорта Kerberos.

Обзор защищенных удаленных команд

В этом разделе описываются защищенные удаленные команды.

Notes:

1. Начиная с распределенной вычислительной среды (DCE) версии 2.2, сервер защиты DCE может возвращать паспорта Kerberos версии 5.
2. Все защищенные удаленные команды (rcmds) используют библиотеку Kerberos версии 5, предоставляемую службой сетевой идентификации (NAS) IBM, которая доступна на DVD-диске пакета расширения. Необходимо установить набор файлов `krb5.client.rte`, который также доступен на DVD-диске пакета расширения.
3. Если миграция операционной системы AIX выполняется с использованием DVD-носителя, а Kerberos уже установлен, то сценарий установки выдаст запрос установки `krb5.client.rte` с DVD-диска пакета расширения.
4. Если миграция операционной системы AIX выполняется с использованием ресурсов NIM, а Kerberos уже установлен, добавьте `krb5` в каталог lpp_source.

Защищенные удаленные команды (rcmds): **rlogin**, **rcp**, **rsh**, **telnet** и **ftp**. Эти команды относят к стандартному способу идентификации AIX. Дополнительные методы относятся к Kerberos.

При использовании идентификации с помощью Kerberos версии 5 клиент получает от сервера защиты DCE или от сервера Kerberos паспорт Kerberos версии 5. Паспорт представляет собой часть текущей среды DCE пользователя или локального одноразового разрешения, зашифрованную для передачи серверу TCP/IP, с которым требуется установить соединение. Демон на сервере TCP/IP расшифровывает паспорт. Такой подход позволяет TCP/IP гарантированно идентифицировать пользователя. Если среде DCE или локальному субъекту, описанному в паспорте, разрешен доступ к учетной записи пользователя в операционной системе,

то запрос на установление соединения обрабатывается. Защищенные удаленные команды поддерживают клиентов Kerberos, а также серверы Kerberos версии 5 и DCE.

Помимо идентификации клиентов Kerberos версии 5 пересылает одноразовое разрешение текущего пользователя серверу TCP/IP. Если одноразовое разрешение помечено как допускающее пересылку, то клиент передает его серверу в качестве начального паспорта Kerberos. Если пользователь обращается к серверу защиты DCE, то на сервере TCP/IP демон преобразует паспорт на выдачу паспорта в полное одноразовое разрешение DCE с помощью команды **k5dcecreds**.

Команда **ftp** применяет способ особый идентификации, отличный от используемого другими защищенными удаленными командами. Эта команда позволяет с помощью механизма защиты GSSAPI обмениваться идентификационными данными между командой **ftp** и демоном **ftpd**. Клиент ftp поддерживает шифрование данных с помощью подкоманд **clear**, **safe** и **private**.

В сеансе связи между клиентом и сервером команда **ftp** позволяет передавать по зашифрованным соединениям данных многобайтовые блоки данных. Стандарт определяет только передачу однобайтовых блоков данных по зашифрованным соединениям. При подключении к системам других производителей с использованием средств шифрования команда **ftp** следует ограничению на передачу однобайтных блоков.

Конфигурация системы:

Для всех защищенных удаленных команд разрешенные в системе способы идентификации определяются настройкой на уровне системы. Конфигурация задается как для входящих, так и для исходящих соединений.

Конфигурация идентификации включает библиотеку `libauthm.a`, а также команды **lsauthent** и **chauthent**, предоставляющие интерфейс командной строки к библиотечным процедурам **get_auth_methods** и **set_auth_methods**.

Способ идентификации определяет, какой способ применяется для идентификации пользователей в сети. Система поддерживает следующие способы:

- Kerberos версии 5 как наиболее распространенный и являющийся основой для DCE.
- Kerberos версии 4, применяемый только защищенными командами rlogin, rsh и rcp. Поддержка этого способа обеспечивается только в системах SP для обратной совместимости с более ранними версиями. Паспорт Kerberos версии 4 не преобразуется в одноразовое разрешение DCE.

Если настроено несколько способов идентификации и с помощью первого способа установить соединение не удалось, то клиент попробует воспользоваться следующим настроенным способом идентификации.

Способы идентификации можно настраивать в любом порядке. Единственное исключение составляет стандартная идентификация AIX, которая должна быть последней в списке методов. Если стандартная идентификация AIX не настроена, то идентификация с помощью пароля не применяется и все попытки подключения с помощью этого способа отклоняются.

Можно также совсем отключить идентификацию. В этом случае система будет отклонять все исходящие и входящие подключения, выполняемые с помощью защищенных удаленных команд. Кроме того, поскольку Kerberos версии 4 поддерживается только командами **rlogin**, **rsh** и **rcp**, то система, настроенная на применение только Kerberos версии 4, будет отклонять запросы на подключение с помощью telnet и FTP.

Идентификация пользователей с помощью Kerberos версии 5:

Метод идентификации Kerberos версии 5 можно использовать для проверки идентификационных данных пользователей.

При использовании идентификации с помощью Kerberos версии 5 клиент TCP/IP получает зашифрованный служебный паспорт для сервера TCP/IP. Когда сервер расшифровывает этот паспорт, он получает доступ к защищенному способу идентификации пользователя (с помощью DCE или локального субъекта). Однако

сервер по-прежнему должен определять, разрешен ли данному DCE или локальному субъекту доступ к локальной учетной записи пользователя. Установление соответствия между DCE или локальным субъектом и учетной записью пользователя операционной системы выполняется общей библиотекой `libvaliduser.a`, а точнее, входящей в ее состав функцией kvalid_user. Если требуется другой способ установления соответствия, то системный администратор должен предусмотреть альтернативу библиотеке `libvaliduser.a`.

Конфигурация DCE:

Для применения защищенных удаленных команд необходимо создать два субъекта DCE для каждого сетевого интерфейса, к которому они могут подключаться.

Субъекты DCE следующие:

```
host/полное_имя_интерфейса
ftp/полное_имя_интерфейса
```

, где *FullInterfaceName* — имя интерфейса и имя домена.

Локальная конфигурация:

Для применения защищенных удаленных команд необходимо создать два локальных субъекта для каждого сетевого интерфейса, к которому они могут подключаться.

Локальные субъекты бывают двух видов:

```
host/полное_имя_интерфейса@имя_области
ftp/полное_имя_интерфейса@имя_области
```

, где *FullInterfaceName* — имя интерфейса и имя домена, а *RealmName* — имя локальной области Kerberos версии 5.

Дополнительная информация по рассматриваемым вопросам приведена в следующих источниках:

- Описание функций get_auth_method и set_auth_method в книге *Technical Reference: Communications, Volume 2*
- Описание команды chauthent в книге *Справочник по командам, том 1*
- Описание команды lsauthent в книге*Справочник по командам, том 3*

Идентификация входа в AIX с помощью службы сетевой идентификации или служб других ОС

В версиях AIX ранее 6.1 загружаемый модуль KRB5 выполнял идентификацию Kerberos через среду Службы сетевой идентификации (NAS), а загружаемый модуль KRB5A выполнял ее через среды других систем. Начиная с AIX версии 6.1 загружаемый модуль KRB5 выполняет идентификацию Kerberos как через среду Службы сетевой идентификации (NAS), так и через среду других систем. Атрибут **is_kadmind_compat** в файле `etc/security/methods.cfg` задает либо среду KRB5, либо среду KRB5A. Начиная с AIX 7.1 загружаемый модуль KRB5A недоступен. Таким образом, атрибут **is_kadmind_compat** следует задавать в файле `etc/security/methods.cfg` для обозначения среды KRB5 или среды KRB5A.

Когда клиент Kerberos настроен на идентификацию через NAS, загружаемый модуль KRB5 осуществляет идентификацию и управление субъектами Kerberos. Этот модуль позволяет администратору системы управлять субъектами Kerberos с помощью команд управления пользователями AIX. Управление субъектами требует от сервера Kerberos поддержки протокола kadmin. NAS обеспечивает эту поддержку через демон **kadmind** (сервер Kerberos, работающий в AIX).

Примечание: Во время настройки клиента Kerberos необходимо включить идентификацию через NAS, иначе клиент будет выполнять идентификацию через службы других ОС, и управление субъектами будет недоступно.

При использовании Kerberos через ОС, отличную от AIX, субъекты Kerberos хранятся в этой ОС, и управление ими через kadmin из AIX невозможно. В этом случае, управление субъектами должно осуществляться отдельно соответствующими инструментам Kerberos. Эти инструменты являются частью конкретной реализации Kerberos или интегрированы в операционную систему (например Windows 2000). Первоначальной целью использования Kerberos через ОС, отличную от AIX, было выполнение идентификации через серверы Windows 2000 Active Directory, где управление субъектами Kerberos осуществляется с помощью инструментов и API управления учетными записями Active Directory. Однако этот подход может использоваться и с другими совместимыми KDC, не поддерживающими интерфейс управления Kerberos.

Установка и настройка интегрированного входа в систему с помощью IBM NAS:

Реализация IBM Kerberos служб сетевой идентификации (NAS) входит в состав пакета расширения.

Чтобы установить пакет сервера Kerberos версии 5, установите набор файлов `krb5.server.rte`:

```
installp -aqXYgd . krb5.server
```

Если система, настраиваемая как сервер Kerberos, одновременно будет и клиентом Kerberos; установите весь пакет KRB5.

Дополнительно, DCE имеет набор утилит для клиентов Kerberos с теми же именами, что и утилиты Kerberos. Во избежание конфликтов имен между командами DCE и Kerberos (например, между командами **klist**, **kinit** и **kdestroy**), команды Kerberos устанавливаются в каталоги `/usr/krb5/bin` и `/usr/krb5/sbin`.

Для запуска команд Kerberos необходимо вводить полное имя исполняемого файла команды. Или добавить каталоги Kerberos в переменную PATH среды.:

```
export PATH=$PATH:/usr/krb5/sbin:/usr/krb5/bin
```

Примечание: Java14 SDK тоже устанавливает команду **kinit**, и она может предшествовать другим командам **kinit** в переменной PATH. Если команды службы сетевой идентификации нужнее команды **kinit** Java14, переместите ее в конец строки, значения переменной PATH.

Документация по службе сетевой идентификации поставляется в пакете `krb5.doc.`*язык*`.pdf|html`, где *язык* указывает на язык документации.

Операционная система AIX обладает двумя базами данных для образования составного загружаемого модуля: LDAP и BUILTIN. Модуль LDAP используется для доступа к информации, хранимой в реестре LDAP (каталоге), а модуль BUILTIN — для доступа к информации, хранимой в реестре файлов (локальной файловой системе). Создаваемый составной загружаемый модуль обычно называется "KRB5files" или "KRB5LDAP". Эти имена показывают, что KRB5 используется либо для идентификации и локальных файлов, либо для LDAP.

Служба сетевой идентификации также поддерживает хранение информации Kerberos либо в локальной файловой системе (устаревшая база данных Kerberos), либо в LDAP. Возможны четыре конфигурации:

- KRB5files с информацией сервера Kerberos, хранящейся в устаревшей базе данных Kerberos.
- KRB5files с информацией сервера Kerberos, хранящейся в базе данных LDAP Kerberos.
- KRB5LDAP с информацией сервера Kerberos, хранящейся в устаревшей базе данных Kerberos.
- KRB5LDAP с информацией сервера Kerberos, хранящейся в базе данных LDAP Kerberos.

Когда для хранения субъектов Kerberos или информации о пользователях и группах AIX используется LDAP, он должен быть настроен до вызова команд настройки Kerberos. После настройки LDAP настройте серверы Kerberos командой **mkkrb5srv**.

Настройка сервера службы сетевой идентификации для работы с хранилищем на основе устаревшей базы данных:

Настройка KDC службы сетевой идентификации и серверов администрирования для работы с устаревшей базой данных Kerberos, а также настройка серверов службы сетевой идентификации, выполняется командой **mkkrb5srv**.

Дополнительная информация об использовании команды **mkkrb5srv** приведена в справке по команде **mkkrb5srv**.

Примечание: Программное обеспечение серверов DCE и Kerberos не рекомендуется устанавливать в одной физической системе одновременно. Если же такая необходимость все-таки возникнет, то необходимо изменить номера портов, применяемые по умолчанию клиентами и сервером DCE или клиентами и сервером Kerberos. В любом случае такое изменение может привести к нарушению взаимодействия с существующими системами, использующими DCE и Kerberos. Сведения о совместном применении DCE и Kerberos приведены в документации по службе сетевой идентификации.

Kerberos версии 5 отклоняет запросы на получение паспортов, поступающие от всех клиентов, часы которых расходятся с часами KDC более чем на указанную величину. По умолчанию максимальное расхождение часов не должно превышать 300 секунд (пять минут). Kerberos требует предварительной настройки какого-либо механизма синхронизации времени между клиентом и сервером. Для синхронизации рекомендуется применять демона **xntpd** или **timed**. Для применения демона **timed** выполните следующие действия:

1. Настройте сервер KDC в качестве сервера времени, запустив на нем демона **timed**:
   ```
   timed -M
   ```
2. Запустите сервер **timed** на каждом клиенте Kerberos:
   ```
   timed -t
   ```
3. Настроить серверы Kerberos KDC и kadmin можно с помощью команды **mkkrb5srv**. Например, для настройки Kerberos в области `MYREALM` с сервером управления `sundial` в домене `xyz.com` нужно будет ввести команду
   ```
   mkkrb5srv -r MYREALM -s sundial.xyz.com -d xyz.com -a admin/admin
   ```

   Подождите несколько минут, пока не будут запущены процессы **kadmind** и **krb5kdc** из файла `/etc/inittab`.

Для хранения информации служба сетевой идентификации использует каталог **/var**. Эта информация включает базу данных, протокол и файлы кэша разрешений идентифицированных пользователей. Размер этих файлов со временем увеличивается. Периодически проверяйте наличие достаточного свободного места в каталоге **/var**.

Обычный вызов команды **mkkrb5srv**:

```
mkkrb5srv -r имя-области -s сервер-KDC -d имя-домена -a имя-администратора
```

Значения переменных в Табл. 16 на стр. 296 используются в следующем примере, показывающем как настроить серверы службы сетевой идентификации для работы с устаревшей базой данных.

Таблица 16. Имена переменных команды **mkkrb5srv**

Имя переменной	Значение переменной
Realm Name	MYREALM
KDC Server	kdcsrv.austin.ibm.com
Domain Name	austin.ibm.com
Administrator Name	admin/admin

Удалите существующую конфигурацию сервера командой **mkkrb5srv –U** или **unconfig.krb5**, если она есть.

Внимание: Если необходимо сохранить имеющуюся конфигурацию сервера Kerberos, не выполняйте следующие действия.

Следующая процедура — пример настройки серверов службы сетевой идентификации для работы с устаревшей базой данных.

1. Введите команду

   ```
   mkkrb5srv -r MYREALM -s kdcsrv.austin.ibm.com -d austin.ibm.com -a admin/admin
   ```

   После ввода этой команды будет предложено ввести главный пароль базы данных.

   Так как служба сетевой идентификации не поддерживает размещение KDC и сервера администрирования на разных хостах, для обоих используется локальный хост. Не обращайте внимание на сообщение об ошибке: `Параметр "-s" не поддерживается`.
2. Когда будет выведен запрос, введите главный пароль базы данных.
3. Когда будет выведен запрос, введите пароль субъекта администратора.

   После ввода пароля субъекта администратора команда **mkkrb5srv** запускает демоны **kadmind** и **krb5kdc** из `/etc/inittab`. Этот процесс может занять несколько минут.
4. Проверьте записи в файле `/etc/inittab`:

   ```
   lsitab krb5kdc
   lsitab kadm
   ```
5. Убедитесь, что серверы KDC и kadmind работают:

   ```
   ps -ef | grep -v grep | grep krb5
   ```

Команда **mkkrb5srv** создает главные административные серверы KDC и kadmind для области Kerberos (MYREALM). Она также создает файлы конфигурации, инициализирует базу данных субъектов и запускает серверы KDC и kadmind.

При запуске команды **mkkrb5srv** выполняются следующие действия:

1. Создается файл `/etc/krb5/krb5.conf`. Имя области, имя сервера управления Kerberos и имя домена берутся из параметров командной строки. В файле `/etc/krb5/krb5.conf` также указываются пути к файлам протоколов default_keytab_name, kdc и admin_server.
2. Создается файл `/var/krb5/krb5kdc/kdc.conf`. В файле `/var/krb5/krb5kdc/kdc.conf` указываются значения переменных *kdc_ports*, *kadmin_port*, *max_life*, *max_renewable_life*, *master_key_type* и *supported_enctypes*. Кроме того, в этом файле задаются пути для переменных *database_name*, *admin_keytab*, *acl_file*, *dict_file* и *key_stash_file*.
3. Создается файл `/var/krb5/krb5kdc/kadm5.acl`. Настраивается управление доступом для субъектов admin, root и host.
4. Создается база данных и один субъект admin. Вам предлагается задать главный ключ Kerberos, а также указать имя и пароль для субъекта администратора Kerberos. Для восстановления после сбоя, главный ключ, а также имя и пароль субъекта администратора, следует хранить в надежном месте.

Дополнительная информация приведена в разделах “Пример выполнения” на стр. 300 и “Сообщения об ошибках и действия по исправлению” на стр. 299.

Настройка сервера Kerberos для использования хранилища LDAP:

kadmin службы сетевой идентификации и серверы KDC для входа в систему с интегрированным Kerberos устанавливаются командой **mkkrb5srv**.

Значения переменных в Табл. 17 используются в следующем примере, показывающем как настроить компоненты серверов службы сетевой идентификации для работы с LDAP.

Таблица 17. Имена переменных команды **mkrb5srv**

Имя переменной	Значение переменной
Realm_Name	MYREALM
KDC_Server	kdcsrv.austin.ibm.com
Domain_Name	austin.ibm.com
Admin_Name	admin/admin
LDAP server	kdcsrv.austin.ibm.com
LDAP administrator name	cn=root
LDAP administrator password	secret

Пример настройки компонентов серверов службы сетевой идентификации для работы с LDAP командой **mkkrb5srv**.

1. Введите следующую команду:

```
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. Убедитесь, что серверы KDC и kadmind работают:

```
ps -ef | grep -v grep | grep krb5
```

Результаты запуска команды **mkkrb5srv** с LDAP похожи на результаты запуска этой команды с устаревшей базой данных. Однако при использовании LDAP, базы данных в файловой системе не создаются. Вместо этого в каталоге `/var/krb5/krb5kdc` создается файл `.kdc_ldap_data` для хранения информации о LDAP.

См. справку по команде **mkkrb5srv**.

Интеграция Kerberos с процедурой входа в систему:

После завершения установки Kerberos необходимо настроить систему на использование Kerberos как основного средства идентификации пользователей.

Для настройки Kerberos в качестве основного средства идентификации запустите команду **mkkrb5clnt** со следующими параметрами:

```
mkkrb5clnt -c KDC -r область -a администратор -s сервер -d домен -A -i база данных -K -T
```

Значения переменных в Табл. 18 на стр. 298 используются в следующем примере, показывающем как настроить в системе вход с идентификацией пользователей через Kerberos, используя файловую систему в качестве хранилища пользователей и групп AIX.

Таблица 18. Имена переменных команды **mkkrb5clnt**

Имя переменной	Значение переменной
Realm Name	MYREALM
KDC Server	kdcsrv.austin.ibm.com
Domain Name	austin.ibm.com
Administration Server	kdcsrv.austin.ibm.com
Administrator Name	admin/admin
База данных пользователей и групп AIX	files

Пример настройки входа в систему с идентификацией пользователей через Kerberos, используя файловую систему в качестве хранилища пользователей и групп AIX.

Введите следующую команду:

```
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
```

При этом будут выполнены следующие действия:

1. Создание файла `/etc/krb5/krb5.conf`. Имя области, имя сервера управления Kerberos и имя домена берутся из параметров командной строки. Пути к файлам протоколов `имя-keytab-по-умолчанию`, `kdc` и `kadmin` тоже обновляются.
2. Параметр **-i** настраивает полностью интегрированный вход в систему. База данных указывает, где хранятся идентификационные данные пользователей AIX. Не следует путать эту базу данных с хранилищем субъектов Kerberos. Хранилище субъектов Kerberos указывается в процессе настройки Kerberos.
3. Флаг **-K** настраивает Kerberos в качестве схемы идентификации по умолчанию. Такой подход позволяет выполнять идентификацию пользователей с помощью Kerberos в момент входа в систему.
4. Флаг **-A** добавляет в базу данных Kerberos запись, делающую пользователя root администратором Kerberos.
5. Флаг **-T** присваивает администратору сервера начальный паспорт на выдачу паспорта.

Примечание: Не применяйте опцию **-D** в команде **mkkrb5clnt** для настройки среды клиента Kerberos для проверки подлинности в Службе сетевой идентификации IBM (NAS). Если вы не укажете опцию **-D** в команде **mkkrb5clnt**, то атрибут **is_kadmind_compat** не будет включен в файл `/usr/lib/security/methods.cfg` и среда клиента Kerberos будет настроена для проверки подлинности в IBM NAS.

Проверьте конфигурацию в файле `/etc/krb5/krb5.conf`. Пример файла `/etc/krb5/krb5.conf` в клиентской системе:

```
[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
```

Примечание: Если в качестве хранилища субъектов Kerberos используется LDAP, в разделе [realms] файла krb5.conf будет содержаться следующая строка:

```
vdb_plugin_lib = /usr/lib/libkrb5ldplug.a
```

Если устанавливаемая система находится в домене DNS, отличном от домена KDC, то необходимо также выполнить следующие дополнительные действия:

1. Изменить файл /etc/krb5/krb5.conf, добавив запись после [domain realm].
2. Установить соответствие между этим доменом и своей областью.

Например, если в область MYREALM нужно добавить клиент из домена abc.xyz.com, измените файл /etc/krb5/krb5.conf следующим образом:

```
[domain realm]
     .austin.ibm.com = MYREALM
     .raleigh.ibm.com = MYREALM
```

После завершения настройки службы сетевой идентификации процедура входа в систему остается неизменной. После входа в систему пользователи будут обладать паспортами на выдачу паспорта, которые будут связываться с запускаемыми процессами. Переменная среды $KRB5CCNAME пользователя указывает на этот паспорт. Чтобы убедиться в том, что вход выполнен успешно, и пользователь имеет паспорт на выдачу паспорта; используйте команду **klist**.

Примечание: При выполнении команды **mkkrb5clnt** следующий раздел добавляется в файл methods.cfg.

```
KRB5:
        program = /usr/lib/security/KRB5
        program_64 = /usr/lib/security/KRB5_64
        options = is_kadmind_compat=yes

KRB5files:
        options = db=BUILTIN,auth=KRB5
```

Более подробная информация о:

- команде **mkkrb5clnt** находится в описании команды **mkkrb5clnt**.
- файле methods.cfg находится в описании файла methods.cfg.

Сообщения об ошибках и действия по исправлению:

При использовании команды **mkkrb5srv** возможны следующие ошибки:

- Если файл krb5.conf, kdc.conf или kadm5.acl уже существует, то команда**mkkrb5srv** не изменяет соответствующие значения. Пользователю выдается сообщение о том, что файл уже существует. При этом любые параметры настройки можно изменить путем непосредственного редактирования файлов krb5.conf, kdc.conf и kadm5.acl.
- Если команда введена неправильно и база данных не создана, то необходимо удалить файлы конфигурации и повторить команду.
- При обнаружении несоответствий между базой данных и значениями конфигурации удалите базу данных из каталога /var/krb5/krb5kdc/* и повторите команду.
- Убедитесь, что в системе запускаются демоны **kadmind** и **krb5kdc**. Проверить работу демонов можно с помощью команды **ps**. Если демоны не запущены, просмотрите файл протокола.

При использовании команды **mkkrb5clnt** возможны следующие ошибки:

- Неправильные значения в krb5.conf можно изменить путем редактирования файла /etc/krb5/krb5.conf.
- Неправильные значения (флаг **-i**) можно исправить путем редактирования файла /usr/lib/security/methods.cfg.

Проверка подлинности вне KRB5 без помощи демона kadmind: Загружаемый модуль KRB5 вызывает задержку, если демон kadmind недоступен и применяется механизм проверки подлинности, отличный от KRB5, например, единый вход в систему (SSO). Эта зависимость устраняется путем задания параметра `kadmind_timeout` в файле **methods.cfg**.

Возможные значения задаются в формате `kadmind_timeout=<время_в_секундах>`, где время_в_секундах должно быть больше нуля.

Когда загружаемый модуль KRB5 пытается подключиться к серверу kadmind, который выключен, возникает тайм-аут Протокола управления передачей (TCP). Параметр `kadmind_timeout` предотвращает дальнейшую задержку после первоначального тайм-аута TCP. Параметр `kadmind_timeout` задает интервал для загружаемого модуля KRB5, в течение которого тот может попытаться установить новое соединение kadmind после первоначального тайм-аута TCP. Если сервер kadmind работает, поведение по умолчанию остается в силе.

По умолчанию параметр `kadmind_timeout` отключен. Для того чтобы включить параметр `kadmind_timeout`, измените файл `methods.cfg` следующим образом:

```
KRB5:
        program = /usr/lib/security/KRB5
        options = kadmind_timeout=300
KRB5files:
        options = db=BUILTIN,auth=KRB5
```

Создаваемые файлы:

Команда **mkkrb5srv** создает следующие файлы:

- `/etc/krb5/krb5.conf`
- `/var/krb5/krb5kdc/kadm5.acl`
- `/var/krb5/krb5kdc/kdc.conf`

Команда **mkkrb5clnt** создает следующие файлы:

- `/etc/krb5/krb5.conf`

Опция **mkkrb5clnt -i** *файлы* добавляет следующий раздел в файл `/usr/lib/security/methods.cfg`:

```
KRB5:
  program =
  options =
KRB5files:
  options =
```

Пример выполнения:

В этом разделе приведен пример выполнения команды.

Ниже приведен пример обработки команды **mkkrb5srv**:

```
# mkkrb5srv -r MYREALM -s sundial.xyz.com -d xyz.com -a admin/admin
```

Вывод команды будет выглядеть примерно следующим образом:

```
  Набор файлов                 Уровень  Состояние  Описание
  ----------------------------------------------------------------------------
Путь: /usr/lib/objrepos
  krb5.server.rte            1.3.0.0  ЗАФИКСИР.  Сервер службы сетевой
                                                 идентификации

Путь: /etc/objrepos
  krb5.server.rte            1.3.0.0  ЗАФИКСИР.  Сервер службы сетевой
                                                 идентификации
```

```
Опция -s не поддерживается.
Сервером управления будет локальный хост.
Инициализация конфигурации...
Создание /etc/krb5/krb5.conf...
Создание /var/krb5/krb5kdc/kdc.conf...
Создание файлов базы данных...
Инициализация базы данных '/var/krb5/krb5kdc/principal' для области 'MYREALM'
Имя главного ключа 'K/M@MYREALM'
Вам будет предложено ввести главный пароль базы данных.
Обязательно запишите и сохраните этот пароль.
Введите главный пароль базы данных:
Еще раз введите главный пароль базы данных для проверки:
Внимание: Не указана стратегия admin/admin@MYREALM;
  по умолчанию без стратегии. Отметим, что
  что стратегия может быть переопределена ограничениями ACL.
Введите пароль для субъекта "admin/admin@MYREALM":
Еще раз введите пароль для субъекта "admin/admin@MYREALM":
Субъект "admin/admin@MYREALM" создан.
Создание таблицы ключей...
Создание /var/krb5/krb5kdc/kadm5.acl...
Запуск krb5kdc...
krb5kdc запущен успешно.
Запуск kadmind...
kadmind запущен успешно.
Команда выполнена успешно.
Перезапуск kadmind и krb5kdc
```

Ниже приведен пример обработки команды **mkkrb5clnt**:

```
mkkrb5clnt -r MYREALM -c sundial.xyz.com -s sundial.xyz.com \
           -a admin/admin -d xyz.com -i files -K -T -A
```

Вывод команды будет выглядеть примерно следующим образом:

```
Инициализация конфигурации...
Создание /etc/krb5/krb5.conf...
Команда выполнена успешно.
Пароль для admin/admin@MYREALM:
Настройка полностью интегрированного входа в систему
Идентификация субъекта admin/admin с существующим удостоверением.
Внимание: Не указана стратегия для host/diana.xyz.com@MYREALM;
  по умолчанию без стратегии. Отметим, что
  что стратегия может быть переопределена ограничениями ACL.
Субъект "host/diana.xyz.com@MYREALM" создан.

Удостоверение администратора не уничтожено.
Идентификация субъекта admin/admin с существующим удостоверением.

Удостоверение администратора не уничтожено.
Идентификация субъекта admin/admin с существующим удостоверением.
Субъект "kadmin/admin@MYREALM" изменен.

Удостоверение администратора не уничтожено.
Настройка Kerberos в качестве схемы идентификации по умолчанию
Настройка root в качестве администратора Kerberos
Идентификация субъекта admin/admin с существующим удостоверением.
Внимание: Не указана стратегия для root/diana.xyz.com@MYREALM;
  по умолчанию без стратегии. Отметим, что
  что стратегия может быть переопределена ограничениями ACL.
Введите пароль для субъекта "root/diana.xyz.com@MYREALM":
Еще раз введите пароль для субъекта "root/diana.xyz.com@MYREALM":
Субъект "root/diana.xyz.com@MYREALM" создан.

Удостоверение администратора не уничтожено.
Очистка удостоверения администратора и выход.
```

Идентификация без применения демона kadmind:

Если демон **kadmind** недоступен, в работе модуля KRB5 может возникнуть сбой. Эта зависимость устраняется параметром *kadmind* файла `methods.cfg`.

Допустимые значения: `kadmind=no`, `kadmind=false` (для выключения обращения к **kadmind**) и `kadmind=yes`, `kadmind=true` (для включения) (значение по умолчанию — "yes"). Если указано значение "no", то демон **kadmind** не участвует в процессе идентификации. Следовательно, состояние демона **kadmind** не учитывается при входе пользователей в систему, и предполагается, что пользователи указывают верные пароли. Однако если этот демон недоступен (например, демон выключен или нет доступа к системе), то администрирование пользователей Kerberos с помощью таких команд AIX, как **mkuser**, **chuser** или **rmuser** будет невозможным.

Значение по умолчанию параметра *kadmind* — `yes`. Это означает, что в ходе идентификации выполняется поиск **kadmind**. Если демон недоступен, то для идентификации может потребоваться больше времени.

Для того чтобы запретить использование демона **kadmind** во время идентификации, внесите в разделы файла `methods.cfg` следующие изменения:

```
KRB5:
        program = /usr/lib/security/KRB5
        options = kadmind=no
KRB5files:
        options = db=BUILTIN,auth=KRB5
```

Если демон **kadmind** недоступен, пользователь root не сможет вносить изменения в пароли пользователей. Например, для изменения пароля, забытого пользователем, потребуется включить демон **kadmind**. Кроме того, если для входа в систему применяется имя субъекта Kerberos, то основное имя субъекта потребуется усечь в соответствии с ограничениями AIX, связанными с длиной имени пользователя. В соответствии с этим усеченным именем извлекается идентификационная информация пользователей AIX (например, имя домашнего каталога).

Если демон **kadmind** недоступен (выключен или недосягаем), команда **mkuser** выдаст следующее сообщение об ошибке:

```
3004-694 Ошибка при добавлении "krb5user": Отсутствуют права доступа.
```

Если параметр *kadmind* имеет значение `no`, или демон **kadmind** недоступен; система не может проверить существование субъекта в базе данных Kerberos, и поэтому не получает атрибуты, связанные с Kerberos. Это приводит к неполным и неточным результатам. Например, команда **lsuser** на запрос ALL может выдавать пустой результат.

К тому же команда **chuser** будет управлять атрибутами AIX, а не Kerberos. Команда **rmuser** не позволит удалять субъекты Kerberos, а команда **passwd** выдаст сообщение об ошибке при обращении к пользователям, идентифицируемым с помощью Kerberos.

Отсутствие доступа к сети, в которой расположен демон **kadmind**, приведет к увеличению времени ожидания ответа. Чтобы избежать задержек, связанных с отсутствием доступа к системе, укажите для параметра kadmind в файле `methods.cfg` значение `no`.

Если демон **kadmind** не работает, пользователи с просроченными паролями не смогут войти в систему или поменять их.

Когда указано `kadmind=no`, а демон **kadmind** работает; можно использовать следующие команды: **login**, **su**, **passwd**, **mkuser**, **chuser** и **rmuser**.

Kerberos через службу сетевой идентификации: информация по устранению неполадок:

Информация по устранению неполадок клиентов Kerberos, использующих сервер Kerberos под управлением операционной системы AIX.

Модуль LDAP выводит сообщения об ошибках и отладочную информацию в подсистему syslog.

Служба сетевой идентификации IBM использует собственные файлы протоколов для записи запросов к демонам KDC и **kadmind**. Имена файлов протоколов указываются в разделе `[logging]` файла `krb5.conf`. Имена по умолчанию: `/var/krb5/log/krb5kdc.log`, `/var/krb5/log/kadmin.log`.

Если неполадка имеет отношение к IBM Tivoli Directory Server, проверьте файлы протокола IBM Tivoli Directory Server. По умолчанию эти файлы имеют имена `/var/ldap/ibmslapd.log` и `/var/ldap/db2cli.log`.

- ***Как в AIX создавать пользователей, идентифицируемых Kerberos?***

  Пользователь "root" должен получить разрешения Kerberos, дающие право выполнять операции по администрированию системы. Эти операции выполняются на сервере KDC kdcsrv.austin.ibm.com.

  Создайте в базе данных Kerberos учетную запись пользователя AIX (foo) и субъект Kerberos (foo@MYREALM):

  ```
  kinit root/kdcsrv.austin.ibm.com
  mkuser -R KRB5files SYSTEM=KRB5files registry=KRB5files foo
  ```

  Также эти команды идентифицируют пользователя, используя файлы KRB5files.

  Если вы настроили LDAP командой **mksecldap**, тогда можно создавать пользователей AIX, идентифицируемых Kerberos, следующей командой:

  ```
  mkuser -R KRB5LDAP SYSTEM=KRB5LDAP registry=KRB5LDAP foo
  ```

- ***Как удалить пользователя, идентифицируемого с помощью Kerberos?***

  Пользователь, идентифицируемый Kerberos, удаляется командой

  ```
  rmuser -R KRB5files foo
  ```

  Если вы настроили LDAP командой **mksecldap**, тогда пользователей, идентифицируемых Kerberos, можно удалять командой

  ```
  rmuser -R KRB5LDAP foo
  ```

- ***Как изменить пароль пользователя, идентифицируемого Kerberos?***

  Пароль пользователя, идентифицируемого Kerberos, изменяется командой

  ```
  passwd -R KRB5files foo
  ```

- ***Каковы расширенные атрибуты Kerberos AIX?***

  Управление информацией о субъекте осуществляется с помощью расширенных атрибутов AIX командами AIX **lsuser** и **chuser**. Только атрибуты с режимом доступа "GET" выводятся на экран. Атрибутам с режимом доступа "SET" привилегированные пользователи могут присваивать значения (в системе AIX это пользователь "root"). Любой пользователь AIX, идентифицируемый Kerberos, может увидеть свои расширенные атрибуты Kerberos и другие разрешенные атрибуты AIX (id, pgrp, groups, gecos, home, shell и пр.).

  Табл. 19 содержит список расширенных атрибутов Kerberos AIX с режимами доступа.

Таблица 19. Расширенные атрибуты Kerberos AIX и их режимы доступа

Имя расширенного атрибута	Описание	Режим доступа
krb5_principal_name	Имя субъекта, связанного с именем пользователя AIX.	GET
krb5_principal	Эквивалентен атрибуту krb5_principal_name.	GET
krb5_realm	Имя области Kerberos, которой принадлежит субъект.	GET
krb5_last_pwd_change	Время последнего изменения пароля субъекта.	GET

Таблица 19. Расширенные атрибуты Kerberos AIX и их режимы доступа (продолжение)

Имя расширенного атрибута	Описание	Режим доступа
krb5_attributes	Набор атрибутов, используемых KDC.	GET/SET
krb5_mod_name	Имя субъекта, который изменял данный субъект последним.	GET
krb5_mod_date	Время последнего изменения субъекта.	GET
krb5_kvno	Версия текущего ключа субъекта (пароля).	GET/SET
krb5_mkvno	Номер версии главного ключа базы данных. Используется для совместимости с другими реализациями. Имеет значение 0.	GET
krb5_max_renewable_life	Максимальный возобновляемый срок действия любого паспорта, выданного этому субъекту.	GET/SET
krb5_names	Список пар имя:имя-хоста. Зарезервирован для будущего использования. Этот атрибут изменять не следует.	GET/SET

Расширенный атрибут krb5_attributes представляет собой набор атрибутов субъекта Kerberos, используемых KDC. Привилегированный пользователь с помощью команды **chuser** может изменять эти атрибуты.

```
chuser -R KRB5files krb5_attributes=+requires_preauth krb5user
```

Чтобы установить флаг, добавьте плюс (+) перед ним. Чтобы сбросить,— минус (-). Например:

+имя-атрибута устанавливает флаг.

-имя-атрибута сбрасывает.

Примечание: При создании пользователя устанавливаются все атрибуты кроме requires_hwauth, needchange, password_changing_service и support_desmd5.

Атрибуты расширенного атрибута krb5_attributes:

allow_postdated
Разрешает выдачу субъекту отсроченных паспортов.

allow_forwardable
Разрешает выдачу переадресуемых паспортов данному субъекту.

allow_tgs_req
Разрешает выдачу служебных паспортов данному субъекту через паспорт на выдачу паспорта.

allow_renewable
Разрешает выдачу возобновляемых паспортов данному субъекту.

allow_proxiable
Разрешает выдачу субъекту proxy-паспортов.

allow_dup_skey
Включает для данного субъекта взаимную идентификацию пользователей.

allow_tix
Разрешает выдачу паспортов данному субъекту.

requires_preauth
Включает обязательную идентификацию программного обеспечения перед выдачей паспорта.

requires_hwauth
Включает обязательную идентификацию аппаратного обеспечения программным перед выдачей паспорта.

needchange
Включает режим обязательной смены ключа (пароля) перед выдачей паспортов данному субъекту.

Примечание: Если этот флаг установлен, пользователю будет предложено сменить пароль при следующем входе в систему. В этом случае, пользователь идентифицируется (с помощью Kerberos), но не получает паспорта на выдачу паспорта. Чтобы получить паспорт на выдачу паспорта, он должен использовать команду **kinit**. Флаг needchange применяется только к Kerberos, использующему модуль "Службы сетевой идентификации".

allow_svr
Разрешает выдачу служебных паспортов данному субъекту.

password_changing_service
Делает данный субъект специальным субъектом для службы изменения паролей.

support_desmd5
Если этот флаг установлен, KDC может выдавать паспорта, использующие алгоритм контрольной суммы MD5 RSA.

Примечание: Установка этого флага может нарушить функциональную совместимость.

- ***Как получить список расширенных атрибутов Kerberos AIX?***

  Чтобы увидеть список расширенных атрибутов Kerberos AIX, используйте команду

  ```
  lsuser -R KRB5files foo
  ```

  Чтобы увидеть конкретные расширенные атрибуты, используйте параметр "-a". Например:

  ```
  lsuser -R KRB5files -f -a krb5_principal krb5_principal_name krb5_realm
  ```

- ***Как изменять расширенные атрибуты Kerberos AIX?***

  Атрибуты: krb5_kvno, krb5_max_renewable_life, krb5_attributes и krb5_names — имеют режим доступа "SET", но изменять их могут только привилегированные пользователи.

  – Установка максимального возобновляемого срока действия паспортов, выдаваемых пользователю foo (5 дней):

    ```
    chuser -R KRB5files krb5_max_renewable_life=432000 foo
    ```

  – Изменение номера версии ключа (пароля) субъекта, связанного с foo:

    ```
    chuser -R KRB5files krb5_kvno=4 foo
    ```

  – Установка всех атрибутов субъекта Kerberos, перечисленных в Табл. 19 на стр. 303:

    ```
    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
    ```

  – Сброс всех атрибутов субъекта Kerberos, перечисленных вТабл. 19 на стр. 303:

    ```
    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
    ```

  – Изменение атрибута krb5_names и добавление пары имя-пользователя/имя-хоста для пользователя AIX:

    ```
    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
    ```

- ***Как получить список всех пользователей, определенных в KRB5files?***

  Чтобы получить список всех пользователей, идентифицируемых Kerberos, введите следующую команду:

  ```
  lsuser -R KRB5files -a registry ALL
  ```

- ***Как преобразовать пользователя AIX в пользователя, идентифицируемого Kerberos?***

  Используйте команду **mkseckrb5** для преобразования пользователя AIX в пользователя, идентифицируемого Kerberos. Команда **mkseckrb5** преобразовывает пользователей-неадминистраторов

(пользователи с ИД больше 201) в пользователей, идентифицируемых Kerberos. При вызове команды **mkseckrb5** будет выведен запрос на ввод имени и пароля субъекта-администратора службы сетевой идентификации. Если не используется параметр "randomize", будет предложено ввести пароль для каждого преобразуемого пользователя.

Примечание: Команда **mkseckrb5** преобразует только локальных пользователей. Пользователей удаленных доменов, таких как LDAP, преобразовывать этой командой нельзя.

Следующий пример *не* использует параметр "randomize" во время преобразования пользователя AIX в пользователя, идентифицируемого Kerberos.

1. Введите следующую команду:

   ```
   mkseckrb5 foo
   ```

2. Перед выполнением входа под этим пользователем, настройте его атрибуты "SYSTEM" и "registry" следующим образом:

   ```
   chuser -R KRB5files SYSTEM=KRB5files registry=KRB5files foo
   ```

Следующий пример использует параметр "randomize" во время преобразования пользователя AIX в пользователя, идентифицируемого Kerberos.

1. Введите следующую команду:

   ```
   mkseckrb5 -r user1
   ```

2. После преобразования задайте атрибуты "SYSTEM", "registry" и пароль пользователя следующим образом:

   ```
   chuser -R KRB5files SYSTEM=KRB5files registry=KRB5files user1
   passwd -R KRB5files user1
   ```

- ***Как изменять пароль субъекта Kerberos?***

  Пользователь "root" может использовать для этого команду **passwd**. Пример:

  ```
  passwd -R KRB5files foo
  ```

  После ввода команды **passwd** показывается следующее сообщение:

  ```
  Изменение пароля для "foo"
  Текущий пароль foo:
  Новый пароль foo:
  Введите новый пароль еще раз:
  ```

  При применении команды **passwd** пользователю "root" текущий пароль вводить не надо. Запрос на ввод текущего пароля можно выключить параметром "rootpwdrequired" в файле `methods.cfg`. Чтобы выключить запрос на ввод текущего пароля, измените файл `/usr/lib/security/methods.cfg` следующим образом:

  ```
  KRB5files:
     options = db=BUILTIN,auth=KRB5,rootrequiresopw=false
  ```

- ***Как после входа в систему получить паспорт на выдачу паспорта, если установлен атрибут "needchange"?***

  Когда установлен флаг "needchange", для получения паспорта на выдачу паспорта после входа в систему используется команда **kinit**. См. атрибут **needhange**.

- ***Почему пароль не принимается AIX?***

  Если пароль не принимается системой, проверьте:

  – работают ли серверы KDC и kadmind;

  – удовлетворяет ли пароль требованиям AIX и службы сетевой идентификации.

- ***Как изменить правила для паролей?***

  Правила для паролей в системе AIX изменяются с помощью атрибутов стратегии управления паролями. Чтобы изменить стратегию управления паролями в базе данных Kerberos, можно воспользоваться инструментом kadmin сервера сетевой идентификации.

- ***Может ли пользователь, идентифицируемый Kerberos, идентифицироваться обычным механизмом идентификации системы AIX?***

Для того чтобы пользователь foo, идентифицируемый Kerberos, идентифицировался функцией **crypt()** системы AIX, сделайте следующее:

1. Задайте командой **passwd** пароль AIX для пользователя foo (`/etc/security/passwd`).
2. Выберите другой пароль для тестирования. Например:

   ```
   passwd -R files foo
   ```
3. Измените атрибут "SYSTEM" этого пользователя следующим образом:

   ```
   chuser -R KRB5files SYSTEM=compat foo
   ```

   Это изменение вместо Kerberos включает идентификацию с помощью функции **crypt()**.

   Примечание: Так как в данном примере пользователь вошел в систему, используя локальную идентификацию; значение AUTHSTATE — compat, паспорт на выдачу паспорта не выдан. Если идентификацию с помощью функции **crypt()** планируется использовать как резервный механизм, перейдите на этап 4.
4. Чтобы использовать идентификацию с помощью функции **crypt()** как резервный механизм, измените атрибут "SYSTEM" следующим образом:

   ```
   chuser -R KRB5files SYSTEM="KRB5files or compat" foo
   ```

- ***Как изменить порт для клиентов kadmind?***

  Демон **kadmind** используется для управления субъектами Kerberos в системах, где применяется идентификация Kerberos, использующих NAS. В следующем примере показано, как изменить порт для клиентов **kadmind**. В этом примере демон **kadmind** работает на сервере `kdcsrv.austin.ibm.com` и использует порт 812.

  1. Используйте команду **config.krb5** для настройки клиента:

     ```
     config.krb5 -C -r MYREALM -c kdcsrv.austin.ibm.com -s \
     kdcsrv.austin.ibm.com -d austin.ibm.com
     ```
  2. Измените номер порта в файле krb5.conf:

     ```
     admin_server = kdcsrv.austin.ibm.com:812
     ```

- ***Как удалить разрешения Kerberos?***

  После каждого входа пользователя в систему предыдущие разрешения Kerberos заменяются. Однако после выхода пользователя из системы они не удаляются. Чтобы их удалить, используйте команду NAS **kdestroy**:

  ```
  /usr/krb5/bin/kdestroy
  ```

- ***Как изменить срок действия паспорта на KDC?***

  Это делается следующим образом:

  1. Измените атрибут "max_life" в файле `kdc.conf`. Например:

     ```
     max_life = 8h 0m 0s
     ```
  2. Перезапустите демоны **krb5kdc** и **kadmind**.
  3. Измените значение "max_life" субъектов krbtgt/MYREALM и kadmin/admin на указанное в 1. Например:

     ```
     kadmin.local
     kadmin.local: modify_principal -maxlife "8 hours" krbtgt/MYREALM
     ```

- ***Что случится, если демон kadmind станет недоступен?***

  Если демон kadmind недоступен, идентификация может занять больше времени, или случится сбой идентификации. Сбой идентификации может возникнуть, если сеть, где расположен демон kadmind, недоступна; или система, в которой работает сервер kadmind, не функционирует. Когда система недоступна, установка параметра kadmind в файле `methods.cfg` в значение `no` устраняет задержки во время идентификации.

  Если демон kadmind выключен, вход в систему с просроченными паролями невозможен. Если демон kadmind недоступен (выключен или недосягаем), и пользователь введет команду **mkuser**; появится следующее сообщение об ошибке:

  ```
  3004-694 Ошибка при добавлении "krb5user": отсутствуют права доступа.
  ```

Помимо этого, команды **chuser** и **lsuser** не смогут управлять атрибутами Kerberos, только атрибутами AIX. Команда **rmuser** не позволит удалять субъекты Kerberos, а команда **passwd** выдаст сообщение об ошибке при обращении к пользователям, идентифицируемыми с помощью Kerberos.

Когда демон kadmind недоступен, пользователь root не может изменять пароли пользователей. Например, для изменения пароля, забытого пользователем, потребуется запустить демон kadmind. Кроме того, если для входа в систему применяется имя субъекта Kerberos, то основное имя субъекта придется укоротить (в соответствии с ограничениями AIX на длину имени пользователя). Укороченное имя используется для получения информации о пользователе AIX (например, для получения имени домашнего каталога пользователя).

- ***Как настроить AIX для входа Kerberos, интегрированного с управлением пользователями AIX и их группами посредством LDAP?***

  Если для хранения информации о пользователях и группах AIX планируется применять LDAP, используйте команду **mksecldap** для настройки сервера и клиента LDAP перед выполнением команд **mkkrb5srv** и **mkkrb5clnt**. Для настройки серверов Kerberos используйте команду **mkkrb5srv**. Для настройки клиента Kerberos используйте команду **mkkrb5clnt** с параметром "–i". Например:

  ```
  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
  ```

- ***Как использовать удаленные команды с поддержкой Kerberos после входа в систему?***

  Когда пользователь AIX идентифицируется в системе с помощью Kerberos, для вызова удаленных команд с поддержкой Kerberos используется паспорт на выдачу паспорта.

  В следующем примере на kdcsrv.austin.ibm.com настраивается сервер NAS с помощью команды **mkkrb5srv**. Командой **mkkrb5clnt** в этой системе также настраивается вход с идентификацией Kerberos. Вторая система, tx3d.austin.ibm.com, настраивается командой **mkkrb5clnt** как клиент.

  1. Сохраните ключи для субъекта хоста, host/tx3d.austin.ibm.com, в файл `/etc/krb5/krb5.keytab` системы tx3d.
  2. Так как для настройки клиентской системы использовалась команда **mkkrb5clnt**, эти ключи извлекаются в файл `/var/krb5/security/keytab/tx3d.austin.ibm.com.keytab`. Создайте ссылку на файл `/etc/krb5/krb5.keytab`:

     ```
     ln -s /var/krb5/security/keytab/tx3d.austin.ibm.com.keytab /etc/krb5/krb5.keytab
     ```

  3. Если система tx3d.austin.ibm.com настраивается сервером Kerberos не под управлением AIX, тогда явно создайте субъект хоста и извлеките ключи. Например:

     ```
     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:
     ```

     Так как инструмент kadmin работает в системе tx3d.austin.ibm.com, ключи извлекаются в файл `/etc/krb5/krb5.keytab` системы tx3d.austin.ibm.com. Это также можно сделать в системе, где расположен сервер Kerberos admin (например kdcsrv). После извлечения ключей в файл "keytab", этот файл передается в систему tx3d и объединяется там с файлом `/etc/krb5/krb5.keytab`.

  4. Настройте удаленные команды, чтобы они использовали идентификацию Kerberos версии 5 системы tx3d.austin.ibm.com:

     ```
     lsauthent
     Standard Aix
     chauthent -k5 -std
     lsauthent
     Kerberos 5
     Standard Aix
     ```

  5. Настройте удаленные команды, чтобы они использовали идентификацию Kerberos версии 5 системы kdcsrv.austin.ibm.com:

     ```
     chauthent -k5 -std
     lsauthent
     Kerberos 5
     Standard Aix
     ```

6. Создайте в системе kdcsrv пользователя (foo), идентифицируемого Kerberos, и задайте его пароль.

```
mkuser -R KRB5files SYSTEM=KRB5files registry=KRB5files foo
passwd -R KRB5files foo
```

7. Создайте пользователя foo в системе tx3d:

```
mkuser -R files foo
```

8. Подключитесь через Telnet к системе kdcsrv.austin.ibm.com, используя идентификацию Kerberos.
9. Создайте паспорт на выдачу паспорта командой **klist**.

```
/usr/krb5/bin/klist
```

Примеры удаленных команд с поддержкой Kerberos.

Примечание: Перед запуском команд примеров удалите файлы `.klogin`, `.rhost` и `hosts.equiv`.

– Введите команду **date** в удаленной системе tx3d.austin.ibm.com с помощью **rsh**:

```
rsh tx3d date
```

– Зайдите в удаленную систему tx3d.austin.ibm.com командой **rlogin**:

```
hostname
kdcsrv.austin.ibm.com
rlogin tx3d -l foo
*******************************
* Вас приветствует AIX Version 6.1! *
*******************************
hostname
tx3d.austin.ibm.com
id
uid=234(foo) gid=1(staff)
```

– Передайте файл в удаленную систему tx3d.austin.ibm.com командой **rcp**:

```
rsh tx3d "ls -l /home/foo"
total 0
echo "Проверка rcp с поддержкой Kerberos" >> 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"
Проверка rcp с поддержкой Kerberos.
```

– Подключитесь через Telnet к системе tx3d.austin.ibm.com, используя разрешения Kerberos:

```
telnet tx3d
Подключение...
Подключение к tx3d.austin.ibm.com выполнено.
Escape-символ - "^]".
[ Kerberos V5 разрешил доступ, ваше имя "foo@MYREALM" ]
```

– Подключитесь через Telnet к системе tx3d.austin.ibm.com и введите имя хоста и ИД:

```
hostname
tx3d.austin.ibm.com
id
uid=234(foo) gid=1(staff)
```

– Перед использованием команды **ftp** с поддержкой Kerberos необходимо создать командой **kadmin** (из системы tx3d.austin.ibm.com) субъект службы FTP `ftp/tx3d.austin.ibm.com` и извлечь его в файл `/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:
```

Пример подключения по FTP к удаленной системе tx3d.austin.ibm.com, используя разрешения Kerberos.

```
ftp tx3d
Имя (tx3d:foo): foo
232 GSSAPI пользователь foo@MYREALM идентифицирован как foo
230-Последний вход: ЦПДВ 17:58:57 ЧТВ 19 МАЯ 2005 по ftp с kdcsrv.austin.ibm.com
230 Пользователь foo вошел в систему.
ftp> ftp> ls -la
```

Настройка клиента Kerberos для работы через сервер Kerberos, управляемый не AIX:

Клиент Kerberos AIX может быть настроен на работу через сервер Kerberos, управляемый ОС Windows Active Directory, Solaris и HP.

Настройка Kerberos на идентификацию через Windows Server Kerberos Service:

Существует несколько способов настройки Kerberos на идентификацию через Windows Server Kerberos Service.

Модуль идентификации Kerberos в KRB5 может использоваться как компонент составного загружаемого модуля. Во время настройки пользователь указывает среду Kerberos для загружаемого модуля. Загружаемый модуль KRB5 — другой метод идентификации через Windows 2000 или Windows 2003 Server Kerberos Service. Псевдозагружаемый модуль AIX BUILTIN обеспечивает доступ к функциям библиотеки защиты. Загружаемый модуль BUILTIN может быть объединен с модулями идентификации для создания компонента базы данных составного загружаемого модуля. Он также предоставляет устаревшее хранилище для пользователей и групп и доступ к файловой системе. Загружаемый модуль LDAP тоже может использоваться в качестве компонента, базы данных, составного загружаемого модуля.

В отличие от других сред Kerberos AIX, работающих через NAS, данная среда не предоставляет управление субъектами Kerberos. Загружаемый модуль KRB5 используется в среде, где субъекты Kerberos хранятся не в системе AIX и не могут управляться этой системой через **kadmin**. Управление субъектами Kerberos осуществляется отдельно специальными инструментами. Эти инструменты являются частью конкретной реализации Kerberos или интегрированы в операционную систему, как сделано в Windows 2000.

Настройка Windows Server 2000 Kerberos Service:

Windows Server 2000 Kerberos Service и клиент NAS совместимы на уровне протокола Kerberos (RFC1510). Поскольку Windows Server 2000 не поддерживает интерфейс **kadmin**, во время настройки клиентов AIX команду **mkkrb5clnt** следует вызвать с параметром "-D". Для управления субъектами в системах Windows следует использовать инструменты этих систем.

Порядок настройки клиента AIX для идентификации с помощью Kerberos через Windows Server 2000 Kerberos Service.

1. Установите Windows Server 2000. Для настройки Microsoft Active Directory Server см. документацию Microsoft.
2. Если клиент NAS не установлен в клиентской системе AIX, установите файл `krb5.client.rte` из пакета расширения AIX.
3. Для настройки клиента Kerberos AIX вызовите команду **mkkrb5clnt**, указав следующую информацию:

   realm Имя домена Windows Active Directory.

   domain
   Имя домена сервера Active Directory.

   KDC Имя хоста сервера Windows.

   server Имя хоста сервера Windows.

   Ниже приведен пример обработки команды **mkkrb5clnt**:

   ```
   mkkrb5clnt -r MYREALM -d austin.ibm.com -c w2k.austin.ibm.com -s w2k.austin.ibm.com -D
   ```

Опция **-D** в команде **mkkrb5clnt** создает опцию **is_kadmind_compat=no** в файле `/etc/methods.cfg` и настраивает среду клиента Kerberos для проверки подлинности в системах, отличных от AIX. Не применяйте опцию **-D** в команде **mkkrb5clnt** для настройки среды клиента Kerberos для проверки подлинности в Службе сетевой идентификации IBM (NAS).

Примечание: При выполнении команды **mkkrb5clnt** следующий раздел добавляется в файл `methods.cfg`.

```
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
```

Более подробная информация о:

- команде **mkkrb5clnt** и допустимых флагах находится в описании команды **mkkrb5clnt**.
- файле `methods.cfg` находится в описании файла `methods.cfg`.

4. Поскольку Windows виды шифрования DES-CBC-MD5 и DES-CBC-CRC, отредактируйте файл `krb5.conf` следующим образом:

```
[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. Создайте субъект хоста.

   Так как имена учетных записей Windows состоят не из нескольких частей, в отличие от имен субъектов NAS; нельзя создать учетную запись, совпадающую с полным именем хоста (`host/`*<полное-имя-хоста>*). Экземпляр субъекта создается с помощью преобразования имени субъекта службы. Для этого создается учетная запись соответствующая субъекту хоста и добавляется преобразование имени субъекта.

   На сервере Active Directory с помощью инструмента Active Directory Management создайте учетную запись пользователя, соответствующую клиентской системе AIX `tx3d.austin.ibm.com`:

   a. Выберите папку `Пользователь`.

   b. Откройте контекстное меню и выберите пункт `Создать`.

   c. Выберите `Пользователь`.

   d. Введите `tx3d` в поле `Имя` и нажмите `Далее`.

   e. Создайте пароль и нажмите `Далее`.

   f. Нажмите `Готово`.

6. В системе Windows Server 2000 введите в консоли команду **Ktpass**, чтобы создать файл `tx3d.keytab` и настроить учетную запись хоста AIX:

```
Ktpass -princ host/tx3d.austin.ibm.com@MYREALM -mapuser tx3d -pass password -out tx3d.keytab
```

7. Скопируйте файл `tx3d.keytab` в систему AIX.

8. В системе AIX добавьте содержимое файла `tx3d.keytab` в файл `/etc/krb5/krb5.keytab`:

```
ktutil
rkt tx3d.keytab
wkt /etc/krb5/krb5.keytab
q
```

9. С помощью инструментов управления пользователями Active Directory создайте учетные записи для домена Windows.

10. Для того чтобы создать учетные записи AIX, соответствующие учетным записям домена Windows, и включить идентификацию Kerberos, выполните следующую команду:

```
mkuser registry=KRB5files SYSTEM=KRB5files foo
```

11. Для того чтобы войти в систему AIX и проверить конфигурацию, используйте команду **telnet**.

Пример сеанса входа в систему, использующего KRB5 и Windows Active Directory:

```
telnet tx3d

Подключение...
Подключение к tx3d.austin.ibm.com выполнено.
Escape-символ - "^]".

telnet (tx3d.austin.ibm.com)
Имя пользователя: foo
Пароль пользователя foo:
*******************************************************************************
* Вас приветствует AIX Version 6.1! *
*******************************************************************************
echo $AUTHSTATE
KRB5files

/usr/krb5/bin/klist
Кэш паспортов: FILE:/var/krb5/security/creds/krb5cc_foo@AUSTIN.IBM.COM_203
Субъект по умолчанию: foo@AUSTIN.IBM.COM

Корректный запуск субъекта службы учета времени действия
29.04.2005 14:37:28 30.04.2004 00:39:22 krbtgt/AUSTIN.IBM.COM@AUSTIN.IBM.COM
     Продление до 30.04.2005 14:37:28

29.04.2005 14:39:22 30.04.2005 00:39:22 host/tx3d.austin.ibm.com@AUSTIN.IBM.COM
```

Настройка Windows Server 2003 Kerberos Service:

Клиент Kerberos может работать через Windows Server 2003 Kerberos Service.

Для того чтобы настроить клиент AIX для работы с Windows Server 2003 Kerberos Service, выполните действия, описанные в “Настройка Windows Server 2000 Kerberos Service” на стр. 310.

Примечание: Команда **kpasswd** клиента NAS не может менять пароли субъектов Kerberos в Windows Server 2003 Kerberos Service. Поэтому после успешного входа в систему AIX, которая использует Kerberos, пользователь не может изменить пароль в Windows Server 2003.

Настройка Kerberos на работу через контроллеры домена Kerberos Sun Solaris и HP-UX.:

Клиент Kerberos можно настроить на работу через контроллеры домена Kerberos Sun Solaris и HP-UX.

В отличие от сред Kerberos AIX, работающих через NAS, данная среда не предоставляет управление субъектами Kerberos. Загружаемый модуль KRB5 используется в среде, где субъекты Kerberos хранятся не в системе AIX и не могут управляться этой системой через **kadmin**. Управление субъектами Kerberos осуществляется отдельно специальными инструментами. Эти инструменты являются частью конкретной реализации Kerberos или интегрированы в операционную систему.

Настройка Sun Solaris:

Настройка клиента Kerberos на идентификацию через систему Sun Solaris.

Sun Enterprise Authentication Mechanism (SEAM) и клиент NAS AIX совместимы на уровне протокола Kerberos (RFC1510). Поскольку интерфейс демона **kadmind** Solaris несовместим с интерфейсом **kadmin** клиента NAS AIX, для настройки клиентов AIX используйте команду **mkkrb5clnt** с параметром "-D". Для управления субъектами в Solaris используйте инструменты этой системы. Так как протоколы изменения паролей у серверов Kerberos SEAM и клиентов NAS AIX различаются, изменение пароля субъекта вызовет сбой настройки.

В следующем примере используется Solaris.

Порядок настройки клиента AIX для идентификации с помощью Kerberos, используя SEAM.

1. Настройте SEAM, следуя документации Sun.
2. Если клиент NAS не установлен в клиентской системе AIX, установите файл `krb5.client.rte` из пакета расширения AIX.
3. Командой **mkkrb5clnt** настройте клиент AIX, используя следующую информацию:

   область
   Имя области Kerberos Solaris: AUSTIN.IBM.COM

   домен Имя домена хоста серверов Kerberos: Austin.ibm.com.

   KDC Имя хоста системы Solaris, в которой работает KDC: sunsys.austin.ibm.com.

   сервер Имя хоста системы Solaris, в которой работает демон **kadmin** (обычно он совпадает с хостом KDC): sunsys.austin.ibm.com.

   Примечание: Так как интерфейсы **kadmin** у Solaris и клиента NAS AIX отличаются, имя сервера не используется клиентами NAS и команду **mkkrb5clnt** необходимо вызывать с параметром "-D".

   Ниже приведен пример обработки команды **mkkrb5clnt**:

   ```
   mkkrb5clnt -r AUSTIN.IBM.COM -d austin.ibm.com\
    -c sunsys.austin.ibm.com -s sunsys.austin.ibm.com -D
   ```

   Опция **-D** в команде **mkkrb5clnt** создает опцию **is_kadmind_compat=no** в файле `/etc/security/methods.cfg` и настраивает среду клиента Kerberos для проверки подлинности в системах, отличных от AIX. Не применяйте опцию **-D** в команде **mkkrb5clnt** для настройки среды клиента Kerberos для проверки подлинности в Службе сетевой идентификации IBM (NAS).

   Примечание: При выполнении команды **mkkrb5clnt** следующий раздел добавляется в файл `methods.cfg`.

   ```
   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
   ```

   Более подробная информация о:

   - команде **mkkrb5clnt** и допустимых флагах находится в описании команды **mkkrb5clnt**.
   - файле `methods.cfg` находится в описании файла `methods.cfg`.
4. С помощью инструмента **kadmin** создайте субъект хоста `host/tx3d.austin.ibm.com@MYREALM` и сохраните его в файл:

   ```
   kadmin: add_principal -randkey host/tx3d.austin.ibm.com
   Создан субъект "host/tx3d.austin.ibm.com@AUSTIN.IBM.COM".

   kadmin:ktadd -k /tmp/tx3d.keytab host/tx3d.austin.ibm.com
   Запись для субъекта host/tx3d.austin.ibm.com с kvno 3,
       тип шифрования DES-CBC-CRC добавлена в файл "keytab" WRFILE:/tmp/tx3d.keytab.

   kadmin: quit
   ```
5. Скопируйте файл `tx3d.keytab` в систему AIX.
6. В системе AIX добавьте содержимое файла `tx3d.keytab` в файл `/etc/krb5/krb5.keytab`:

   ```
   ktutil
   rkt tx3d.keytab
   l
   Замок  KVNO   Субъект
   wkt /etc/krb5/krb5.keytab
   q
   ```
7. Создайте субъект Kerberos инструментом Solaris **kadmin**.

   ```
   add_principal sunuser
   ```

8. Создайте учетные записи AIX, соответствующие субъекту Kerberos Solaris, и включите идентификацию Kerberos:

   ```
   mkuser registry=KRB5files SYSTEM=KRB5files sunuser
   ```

9. С помощью команды **telnet** войдите в систему AIX под именем sunuser и проверьте конфигурацию.

   Пример сеанса входа в систему, использующего KRB5 и KDC Solaris:

   ```
   telnet tx3d

   echo $AUTHSTATE
   KRB5files

   echo $KRB5CCNAME
   FILE:/var/krb5/security/creds/krb5cc_sunuser@AUSTIN.IBM.COM_207

   Посмотрите разрешения:
   /usr/krb5/bin/klist
   ```

Настройка HP-UX:

Настройка клиента Kerberos на идентификацию через систему HP-UX.

Процедура идентификации через систему HP-UX 11i аналогична описанной в “Настройка Sun Solaris” на стр. 312. KDC HP-UX и клиент NAS AIX совместимы на уровне протокола Kerberos (RFC1510). Протокол изменения паролей также совместим. Поскольку интерфейс демона **kadmind** HP-UX несовместим с интерфейсом **kadmin** клиента NAS AIX, для настройки клиентов AIX следует использовать команду **mkkrb5clnt** с параметром "-D".

Порядок настройки клиента AIX для идентификации с помощью Kerberos, используя HP-UX 11i Kerberos версии 2.1.

1. Настройте HP-UX 11i Kerberos версии 2.1, следуя документации HP.
2. Если клиент NAS не установлен в клиентской системе AIX, установите файл `krb5.client.rte` из пакета расширения AIX.
3. Для настройки клиента Kerberos AIX вызовите команду **mkkrb5clnt**, указав следующую информацию:

   область
   Имя области HP Kerberos: HPSYS.AUSTIN.IBM.COM.

   домен Имя домена хоста серверов Kerberos HP-UX: austin.ibm.com.

   KDC Имя хоста системы HP-UX, в которой работает KDC: hpsys.austin.ibm.com.

   сервер Имя хоста сервера HP-UX: hpsys.austin.ibm.com.

   Примечание: Так как интерфейсы **kadmin** у HP-UX и клиента NAS AIX отличаются, имя сервера не используется клиентами NAS и команду **mkkrb5clnt** необходимо вызывать с параметром "-D".

   Ниже приведен пример обработки команды **mkkrb5clnt**:

   ```
   mkkrb5clnt -r AUSTIN.IBM.COM -d austin.ibm.com\
    -c hpsys.austin.ibm.com -s hpsys.austin.ibm.com -D
   ```

   Опция **-D** в команде **mkkrb5clnt** создает опцию **is_kadmind_compat=no** в файле `/etc/security/methods.cfg` и настраивает среду клиента Kerberos для проверки подлинности в системах, отличных от AIX. Не применяйте опцию **-D** в команде **mkkrb5clnt** для настройки среды клиента Kerberos для проверки подлинности в Службе сетевой идентификации IBM (NAS).

   Примечание: При выполнении команды **mkkrb5clnt** следующий раздел добавляется в файл `methods.cfg`.

   ```
   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
```

Более подробная информация о:

- команде **mkkrb5clnt** и допустимых флагах находится в описании команды **mkkrb5clnt**.
- файле `methods.cfg` находится в описании файла `methods.cfg`.

4. В файле `krb5.conf` поменяйте тип шифрования на тот, что был указан во время установки Kerberos HP-UX (**krbsetup**). Если используется DES-CRC, отредактируйте раздел [libdefaults] файла `krb5.conf` клиента AIX следующим образом:

   ```
   default_tkt_enctypes = des-cbc-crc

   default_tgs_enctypes = des-cbc-crc
   ```

5. С помощью инструмента HP-UX **kadmin_ui** создайте субъект хоста `host/tx3d.austin.ibm.com`.
6. Сохраните ключ в файле. Для этого в меню `Правка` окна `Информация о субъекте` выберите команду `Извлечь ключ службы`.
7. Скопируйте файл `tx3d.keytab` в систему AIX.
8. В системе AIX добавьте содержимое файла `tx3d.keytab` в файл `/etc/krb5/krb5.keytab`:

   ```
   ktutil
   rkt tx3d.keytab
   l
   Замок  KVNO   Субъект
   wkt /etc/krb5/krb5.keytab
   q
   ```

9. С помощью инструмента HP-UX **kadmin_ui** создайте субъект Kerberos "hpuser" и сбросьте флаг pw_require на вкладке `Изменить/Атрибут`.
10. Создайте учетную запись AIX, соответствующую субъекту Kerberos в HP-UX:

    ```
    mkuser registry=KRB5files SYSTEM=KRB5files hpuser
    ```

11. С помощью команды **telnet** войдите в систему AIX под именем hpuser и проверьте конфигурацию.

    Пример сеанса входа в систему, использующего KRB5 для идентификации через систему HP-UX:

    ```
    telnet tx3d

    echo $AUTHSTATE
    KRB5files

    Посмотрите разрешения:
    /usr/krb5/bin/klist
    ```

12. Смените пароль командой **passwd**.

    Примечание: При смене пароля применяется стратегия управления паролями HP-UX. Как настроить стратегию управления паролями, см. документацию по HP-UX.

Kerberos через другие ОС: вопросы и информация по устранению неполадок:

Этот раздел содержит ответы на вопросы о клиентах Kerberos, использующих сервер Kerberos под управлением не ОС AIX.

Примечание: В следующих примерах используется Microsoft Active Directory Server. Но эти примеры также применимы к операционным системам Solaris и HP.

Приступая к устранению неполадок, первым делом следует убедиться, что все серверы и демоны работают.

В системах, отличных от AIX, Kerberos использует подсистему syslog для вывода сообщений об ошибках и информации для отладки. См. документацию по демону **syslogd**.

- ***Как создать пользователя AIX?***

Учетная запись пользователя (foo) AIX создается командой

```
mkuser registry=KRB5files SYSTEM=KRB5files foo
```

Команда **mkuser** создает пользователя в системе AIX. Кроме этого, необходимо создать учетную запись пользователя в Windows Server Active Directory, соответствующую учетной записи AIX. При создании учетной записи пользователя в Windows Server Active Directory неявно создаются субъекты.

- ***Как удалить пользователя, идентифицируемого с помощью Kerberos?***

  Пользователь, идентифицируемый с помощью Kerberos, удаляется командой

  ```
  rmuser -R KRB5files foo
  ```

  Команда **rmuser** удаляет пользователя из системы AIX. Кроме этого, необходимо удалить соответствующую учетную запись из Windows Server Active Directory с помощью инструментов управления пользователями Windows Server.

- ***Как изменить пароль пользователя, идентифицируемого Kerberos?***

  Пароль пользователя, идентифицируемого Kerberos, изменяется командой

  ```
  passwd -R KRB5files foo
  ```

  Если KDC поддерживает команду **kpasswd**, команда **passwd** изменяет пароль субъекта Kerberos foo@MYREALM на сервере Kerberos.

- ***Как разрешить пользователям изменять на клиенте просроченные пароли?***

  Для этого нужно добавить параметр allow_expired_pwd=yes в файл `methods.cfg`. Когда параметр allow_expired_pwd имеет значение `yes`, пользователям с просроченными паролями будет предложено их изменить. Если этот параметр имеет значение `no` или `отсутствует` в файле, такие пользователи не смогут пройти идентификацию.

  ```
  KRB5:
     program = /usr/lib/security/KRB5
     options = authonly,allow_expired_pwd=yes
  ```

- ***Как преобразовать пользователя AIX в пользователя, идентифицируемого Kerberos?***

  Преобразование пользователя AIX в пользователя, идентифицируемого с помощью Kerberos, делается следующим образом:

  1. Проверьте наличие учетной записи пользователя в Windows Server Active Directory командой

     ```
     chuser registry=KRB5files SYSTEM=KRB5files foo
     ```

  2. Если у пользователя нет учетной записи в Active Directory, создайте ее и установите атрибуты "SYSTEM" и "registry" командой **chuser**. Имя пользователя в Active Directory может отличаться от соответствующего имени пользователя AIX. Если имя пользователя AIX отличается, необходимо использовать атрибут auth_name для преобразования этого имени в имя Active Directory.

     ```
     chuser registry=KRB5files SYSTEM=KRB5files auth_name=Christopher chris
     ```

- ***Что делать, если не удается вспомнить пароль?***

  Если пароль забыт, администратором Active Directory должен быть задан новый пароль. Пользователь root AIX не может устанавливать пароли субъектов Kerberos Active Directory.

- ***Для чего предназначены атрибуты auth_name и auth_domain?***

  Примечание: Это необязательные атрибуты. Если система AIX поддерживает имена пользователей длиной более 8 символов, атрибут auth_name может оказаться ненужным.

  Атрибуты auth_name и auth_domain преобразуют имена пользователей AIX в имена субъектов Kerberos KDC. Например, если имя пользователя AIX, chris, имеет атрибуты auth_name=christopher и auth_domain=SOMEREALM; то имя субъекта Kerberos будет christopher@SOMEREALM. При использовании атрибута auth_domain запросы посылаются в область SOMEREALM вместо области по умолчанию. Это позволяет пользователю chris идентифицироваться в области SOMEREALM, а не в MYREALM. В данном примере в файл `krb5.conf` необходимо добавить имя области SOMEREALM.

- ***Может ли пользователь, идентифицируемый с помощью Kerberos, идентифицироваться через обычный механизм идентификации AIX?***

  Да. Это делается следующим образом:

1. Задайте пароль AIX (/etc/security/passwd) командой **passwd**:

   ```
   passwd -R files foo
   ```

2. Измените атрибуты "registry" и "SYSTEM":

   ```
   chuser -R KRB5files registry=files SYSTEM=compat foo
   ```

   Эта команда меняет метод идентификации с Kerberos на compat (который использует системную функцию "crypt()"). При следующем входе пользователя foo в систему будет использоваться локальный пароль из файла /etc/security/passwd.

   Идентификацию с помощью функции crypt() можно использовать как резервный механизм идентификации на случай сбоя Kerberos. Для этого необходимо изменить атрибут "SYSTEM":

   ```
   chuser -R KRB5files SYSTEM="KRB5files or compat" foo
   ```

- ***Нужно ли устанавливать сервер Kerberos в AIX, если используется Windows Server 2000 Kerberos Service?***

  Нет, т. к. пользователи идентифицируются KDC Active Directory. Если в качестве сервера Kerberos для других целей планируется использование KDC службы сетевой идентификации AIX, то установка сервера необходима.

- ***Что делать, если AIX не распознает пароль?***

  Если AIX не распознает пароль, выполните следующие действия:

  – Проверьте связь с Windows 2000 Active Directory Server.

  – Убедитесь, что пароль удовлетворяет требованиям AIX и Windows Server 2000 Active Directory. Информация об изменении правил стратегии паролей в AIX приведена в разделе Изменение стратегии показа.

  Примечание: Пароль Windows Server 2003 Kerberos Service изменить нельзя.

- ***Что делать, если не удается войти в систему?***

  Если вам не удалось войти в систему, выполните следующие действия:

  – В системе Windows проверьте, работает ли KDC:

    1. В "Панели управления" откройте "Администрирование".
    2. Выберите "Службы".
    3. Убедитесь, что служба "Центр распределения ключей Kerberos" работает.

  – В системе AIX проверьте параметры в файле /etc/krb5/krb5.conf. Они должны быть корректными и указывать на правильный KDC.

  – В системе AIX проверьте наличие в файле "client-keytab" ключа хоста. Например, если файл "keytab" по умолчанию — /etc/krb5/krb5.keytab, выполните следующую команду:

    ```
    ktutil
    rkt /etc/krb5/krb5.keytab
    l
    ```

  – Убедитесь, что выходные данные команды **kvno** в файле "keytab" соответствуют выходным данным команды **Ktpass**.

  – Проверьте значение атрибутов auth_name и auth_domain (если установлены). Они должны ссылаться на правильное имя субъекта KDC Active Directory.

  – Убедитесь, что в атрибуте "SYSTEM" указан вход в систему с помощью Kerberos.

  – Проверьте, не просрочен ли пароль.

- ***Как выключить проверку паспорта на выдачу паспорта?***

  Проверка выключается параметром файла /usr/lib/security/methods.cfg в разделе KRB5:

  ```
  KRB5:
     program = /usr/lib/security/KRB5
     options = tgt_verify=no
  KRB5files:
     options = db=BUILTIN,auth=KRB5
  ```

  Допустимые значения параметра "tgt_verify": no, false (для выключения проверки) и yes, true (для включения). По умолчанию проверка паспорта на выдачу паспорта выполняется. Когда параметр

"tgt_verify" имеет значение no, проверка паспорта на выдачу паспорта не выполняется, и передача ключей субъектов хоста не нужна. Сделанные изменения исключают обращение к файлу "keytab" только во время выполнения идентификации. Другие приложения с поддержкой Kerberos могут требовать файл keytab для субъектов хоста и службы.

- ***Что делать, если вход в систему невозможен, поскольку преобразование имен хостов не работает, и полные имена хостов не распознаются?***

  Проверка паспорта на выдачу паспорта требует, чтобы субъект host/<имя-хоста> создавался в KDC. Это имя хоста — полное имя клиента в системе, где выполняется идентификация. Клиентская система запрашивает паспорт службы, используя имя субъекта хоста — host/<имя-хоста>. В некоторых конфигурациях клиентская система не может получить полное имя хоста и вместо него получает короткое имя. В таких случаях, возникает несоответствие — проверка паспорта на выдачу паспорта приводит к неудовлетворительному результату — вход в систему не удается. Например, если файл /etc/hosts содержит только короткое имя, а в файле /etc/netsvc.conf указано hosts=local,bind; процедура преобразования имени вернет короткое имя.

  Неполадки преобразования имен устраняются следующим образом:

  – Измените порядок преобразования имен в файле /etc/netsvc.conf, чтобы возвращалось полное имя хоста. Файл netsvc.conf определяет порядок преобразования имен хостов и псевдонимов.

    В следующем примере преобразователь имен использует службу BIND для преобразования имени хоста. Если служба BIND не может преобразовать имя, используется файл /etc/hosts. Если оба метода не дают результатов, используется NIS.

    ```
    hosts=bind,local,nis
    ```

    Если первым методом преобразования должен быть local, поменяйте короткое имя хоста (myhost) в файле /etc/hosts на полное (myhost.austin.ibm.com).

  – Если проверка паспорта на выдачу паспорта не требуется, ее можно выключить. См. *Как выключить проверку паспорта на выдачу паспорта?*.

- ***Почему процедура passwdexpired возвращает 0, когда срок действия пароля пользователя kerberos истек на сервере kerberos, отличном от AIX?***

  Процедура **passwdexpired** возвращает 0, так как информацию об истечении срока действия пароля нельзя получить напрямую от сервера kerberos, отличного от AIX, из-за несовместимости или отсутствия интерфейсов **kadmin**.

  Флаг **allow_expired_pwd** в файле **methods.cfg** позволяет AIX получить информацию об истечении срока действия пароля с помощью интерфейсов проверки подлинности kerberos. Фактическую информацию об истечении срока действия пароля можно получить во время входа в систему или путем вызова процедур **authenticate** и **passwdexpired**.

Модуль Kerberos

Модуль Kerberos является расширением ядра, которое используется клиентом и сервером NFS. С его помощью клиент и сервер NFS обрабатывают функции Kerberos, обеспечивающие целостность и секретность сообщений, не вызывая демон **gss**.

Модуль Kerberos загружается демоном **gss**. Используемые методы основаны на Службе сетевой идентификации версии 1.2, которая в свою очередь основана на MIT Kerberos.

Расположение модуля Kerberos: /usr/lib/drivers/krb5.ext.

Ознакомьтесь также с описанием демона **gss**.

Информация, связанная с данной:

Ресурсы IBM developerWorks в IBM Network Authentication Service и связанных технологиях для AIX

Сервер RADIUS

Служба IBM RADIUS представляет собой протокол сетевого доступа, предназначенный для идентификации, предоставления прав доступа и учета. Это протокол на основе портов, управляющий взаимодействием между Серверами сетевого доступа (NAS), а также серверами идентификации и учета.

NAS работает в качестве клиента RADIUS. Для идентификации транзакций между клиентом и сервером RADIUS применяется *общий шифр*, который не передается по сети. Кроме того, шифрование применяется для защиты паролей, передаваемых между клиентом и сервером RADIUS.

Клиент отвечает за передачу сведений о пользователе соответствующим серверам RADIUS и обработку ответов, возвращаемых этими серверами. Серверы RADIUS получают запросы на установление соединений, выполняют идентификацию пользователей и возвращают информацию о конфигурации, необходимую клиенту для предоставления службы пользователю. Сервер RADIUS может выполнять роль клиента Proxy для других серверов RADIUS, если конфигурация настроена соответствующим образом. В качестве транспортного протокола сервер RADIUS применяет протокол пользовательских дейтаграмм (**UDP**).

Протокол идентификации и предоставления прав доступа, применяемый сервером RADIUS, соответствует стандарту IETF RFC 2865. Кроме того, сервер поддерживает протокол учета, описанный в RFC 2866. Прочие поддерживаемые стандарты: RFC 2284 (EAP), RFC 2869 (частично), сообщения об истечении срока действия паролей RFC 2882, MD5-Challenge и TLS. Более подробная информация об этих RFC приведена в следующих разделах:

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

Информацию обо всех стандартах RFC можно найти на web-сайте http://www.ietf.org.

Установка сервера RADIUS

Сервер RADIUS можно установить с помощью программы SMIT или команды **installp**. Программное обеспечение RADIUS поставляется на основном установочном носителе AIX. Имена соответствующих образов следующие: radius.base и bos.msg.<язык>.rte.

Если вы планируете хранить имена пользователей и пароли в каталоге LDAP, то дополнительно потребуется установить образ ldap.server. Программное обеспечение **installp** следует устанавливать на каждом экземпляре сервера RADIUS.

Если планируется использовать идентификацию EAP-TLS (например, для идентификации цифровых сертификатов в беспроводной сети), необходимо установить OpenSSL версии не ниже 0.9.7 и в файле `/etc/radius/radiusd.conf` указать полный путь к библиотеке libssl.a.

Демоны RADIUS запускаются командой **radiusctl**. Во время работы в системе выполняются несколько процессов radiusd, по одному на:

- идентификацию,
- учет,
- наблюдение за другими демонами.

Во время загрузки системы демоны автоматически запускаются на уровне выполнения 2, если RADIUS не настроен на использование EAP-TLS.

Чтобы изменить это поведение, отредактируйте файл /etc/rc.d/rc2.d/Sradiusd.

Примечание: Если RADIUS настроен на идентификацию цифровых сертификатов с помощью EAP-TLS, демоны не должны запускаться автоматически, т. к. пароль сертификата вводится администратором. В этом случае, RADIUS необходимо запускать и перезапускать вручную командой **radiusctl**.

Останов и перезапуск сервера RADIUS

Каждый раз при внесении изменений в файл конфигурации сервера RADIUS /etc/radius/radiusd.conf, либо в файлы предоставления прав доступа по умолчанию /etc/radius/authorization/default.policy и /etc/radius/authorization/default.auth необходимо перезапустить демоны **radiusd**. Это делается из SMIT или командной строки.

Команды запуска, перезапуска и остановки сервера RADIUS:

```
radiusctl start
radiusctl restart
radiusctl stop
```

Останов и запуск сервера RADIUS необходим, поскольку демон должен создать таблицу памяти со всеми атрибутами по умолчанию, указанными в файлах конфигурации, описанных выше. Общая память применяется для каждого локального пользователя. Локальная таблица пользователей по причинам, связанным с производительностью, создается только в процессе инициализации демона.

Функция On-demand:

При необходимости вы можете запустить несколько демонов идентификации и учета RADIUS.

Каждому серверу для приема запросов выделяется отдельный порт. В файле radiusd.conf по умолчанию для идентификации указан порт 1812, для учета - 1813. Данные номера портов соответствуют стандарту IANA. По своему усмотрению вы можете добавить в файл radiusd.conf дополнительные порты. Перед добавлением номера порта убедитесь, что он еще не выделен другим службам. Если в полях **Authentication_Ports** и **Accounting_Ports** файла radiusd.conf указано несколько значений, то для каждого запускается отдельный экземпляр демона **radiusd**. Демоны принимают запросы, поступающие на соответствующие порты.

Файлы конфигурации RADIUS

Демон RADIUS использует несколько файлов конфигурации. Примеры этих файлов входят в комплект поставки службы RADIUS.

Все файлы конфигурации принадлежат пользователю root и группе security. Все файлы конфигурации, за исключением словаря, можно редактировать с помощью программы (SMIT) - инструмента управления системой. Изменения, внесенные в файлы конфигурации, вступают в силу после перезапуска сервера.

Файл radiusd.conf:

В файле radiusd.conf содержатся параметры настройки RADIUS.

По умолчанию RADIUS ищет файл radiusd.conf в каталоге /etc/radius. Записи файла конфигурации должны соответствовать приведенному формату. RADIUS принимает только допустимые ключевые слова и значения, в противном случае применяются значения по умолчанию. После запуска демонов RADIUS проверьте наличие сообщений об ошибках параметров конфигурации в файле SYSLOG. Некоторые ошибки не приводят к останову сервера.

Этот файл следует защитить от чтения и записи, поскольку он влияет на работу серверов идентификации и учета. Кроме того, в этом файле могут содержаться конфиденциальные данные.

Важное замечание: При редактировании файла `radiusd.conf` не изменяйте существующий порядок записей. Порядок записей учитывается на панелях программы SMIT.

Ниже приведен пример файла `radiusd.conf`:

```
#------------------------------------------------------------------#
#               ФАЙЛ КОНФИГУРАЦИИ                                  #
#                                                                  #
#  По умолчанию RADIUS будет искать radiusd.conf в каталоге        #
#  /etc/radius.                                          #
#                                                                  #
#  Записи файла конфигурации должны соответствовать приведенным    #
#  форматам.  RADIUS принимает только допустимые пары              #
#  "Ключевое слово : значение(я)". Если пара ошибочна или          #
#  отсутствует, будет применяться значение по умолчанию.                                    #
#                                                                  #
#  При запуске демона RADIUS проверьте сообщения об ошибках        #
#  параметров конфигурации в файле SYSLOG.                         #
# Опять же, не все ошибки могут привести к остановке            #
#  сервера.                                             #
#                                                                  #
#  И наконец, этот файл должен быть защищен от чтения и записи,    #
#  поскольку он влияет на работу серверов идентификации            #
#  и учета. Кроме того, в этом файле могут содержаться             #
#  конфиденциальные данные.                                    #
#                                                                  #
#  ПРИ ИЗМЕНЕНИИ ЭТОГО ФАЙЛА НЕ ИЗМЕНЯЙТЕ ПОРЯДОК ЗАПИСЕЙ          #
#  В НЕМ. ОТ ЭТОГО ПОРЯДКА ЗАВИСЯТ ПАНЕЛИ SMIT.          #
#                                                                  #
#                                                                  #
#------------------------------------------------------------------#

#------------------------------------------------------------------#
#               Глобальная конфигурация                            #
#                                                                  #
#   RADIUSdirectory  :  Базовый каталог демона RADIUS, в котором   #
#                       должны быть расположены прочие файлы       #
#                       конфигурации.                              #
#                                                                  #
#   Database_location : Задает применяемый способ идентификации.   #
#                       Допустимые значения: Local, LDAP, UNIX     #
#                       UNIX  - Пользователи системы AIX           #
#                       Local - Локальная база данных AVL с raddbm #
#                       LDAP  - Центральная база данных            #
#                                                                  #
#   Local_Database   :  Задает имя локальной базы данных.          #
#                       Этот флаг должен быть задан, если для      #
#                       флага Database_location указано значение   #
#                       Local.                                     #
#                                                                  #
#   Debug_Level      :  Задает уровень отладки для сервера RADIUS. #
#                       Допустимые значения: 0, 3 и 9.             #
#                       По умолчанию применяется значение 3.       #
#                       Вывод отправляется в расположение,         #
#                       указанное в разделе *.debug в файле        #
#                       /etc/syslog.conf                           #
#                                                                  #
#                       С повышением уровня увеличивается число    #
#                       сообщений, отправляемых в файл syslog.     #
#                       Например, уровень "9" кроме собственных    #
#                       сообщений предусматривает отправку         #
#                       сообщений, создаваемых на уровнях          #
#                       "0" и "3".                                 #
```

```
#                                                                     #
#                       0 : минимальный уровень детализации           #
#                           протокола syslogd.  Для каждого            #
#                           процесса RADIUS заносятся сообщения о     #
#                           его начале и окончании.  Также заносятся#
#                           условия ошибок.                                 #
#                                                                     #
#                       3 : заносятся сообщения ACCESS ACCEPT,        #
#                           REJECT и DISCARD для каждого пакета.      #
#                           Этот уровень обеспечивает ведение         #
#                           контрольного журнала идентификации.       #
#                                                                     #
#                       9 : Максимальная степень детализации. Ука- #
#                           зываются значения атрибутов при           #
#                           обработке транзакций, и так               #
#                           далее.                    #
#                      [Не рекомендуется в обычном режиме работы]  #
#                                                                     #
#---------------------------------------------------------------------#
RADIUSdirectory   : /etc/radius
Database_location : UNIX
Local_Database    : dbdata.bin
Debug_Level       : 3
#---------------------------------------------------------------------#
#                 Конфигурация учета                                  #
#                                                                     #
#  Local_Accounting :  Если значение этого флага равно ON или TRUE, #
#                      то файл будет содержать запись пакетов       #
#                      ACCOUNTING START и STOP, полученных от       #
#                      сервера NAS.  Файл протокола по умолчанию:   #
#                                                                     #
#                          /var/radius/data/accounting              #
#                                                                     #
#  Local_accounting_loc: полное имя файла данных локального         #
#                        учета /var/radius/data/accounting.         #
# Применяется только если Local_             #
#                        Accounting=ON. Если значение по            #
#                        умолчанию изменено, то каталог и файл      #
#                        (с соответствующими правами доступа)       #
#                        должен создать администратор.                          #
#                                                                     #
#---------------------------------------------------------------------#
Local_Accounting  : ON
Local_Accounting_loc : /var/radius/data/accounting
#---------------------------------------------------------------------#
#     Атрибуты ответных сообщений                                     #
#                                                                     #
#     Accept_Reply-Message : Отправляется, если сервер RADIUS       #
#                            возвращает пакет Access-Accept           #
#                                                                     #
#     Reject_Reply-Message : Отправляется, если сервер RADIUS       #
#                            возвращает пакет Access-Reject           #
#                                                                     #
#     Challenge_Reply-Message : Отправляется, если сервер RADIUS    #
#                               возвращает пакет Access-Challenge   #
#---------------------------------------------------------------------#
Accept_Reply-Message :
Reject_Reply-Message :
Challenge_Reply-Message :
Password_Expired_Reply-Message :
#---------------------------------------------------------------------#
#    Продление просроченных паролей                                   #
#                                                                     #
#    Allow_Password_Renewal: YES или NO                               #
#                            Если указано значение YES,               #
#                            пользователи могут обновлять             #
#                            просроченные пароли с помощью            #
```

```
#                           протокола RADIUS. Для этого требуется #
#                           аппаратная поддержка пакетов          #
#                           Access-Password-Request.              #
#-----------------------------------------------------------------#
Allow_Password_Renewal  : NO
#-----------------------------------------------------------------#
#    Идентификатор сообщения в пакете Access-Request              #
#                                                                 #
#    Require_Message_Authenticator: YES или NO                    #
#                                   Если указано значение YES,    #
#                                   проверяется наличие           #
#                                   идентификатора сообщения в    #
#                                   пакете Access-Request. Если   #
#                                   идентификатор отсутствует,    #
#                                   пакет будет отброшен.         #
#-----------------------------------------------------------------#
Require_Message_Authenticator : NO
#-----------------------------------------------------------------#
#       Серверы (идентификация и учет)                            #
#                                                                 #
#   Authentication_Ports : Указывает порты, применяемые для приема #
#                          запросов серверами идентификации. Если #
#                          значение в этом поле не указано, демон #
#                          идентификации запущен не будет.        #
#                          В этом поле можно указать несколько    #
#                          значений, перечисленных через          #
#                          запятую ','.                           #
#                                                                 #
#                          Допустимы только числовые значения,    #
#                          например "6666". В этом случае демон   #
#                          сервера будет принимать запросы на порт #
#                          "6666".                                #
#                                                                 #
#   Accounting_Ports     : Аналогично полю Authentication_Ports.  #
#                          См. определение, приведенное выше.     #
#                                                                 #
# [Примечание] Проверка конфликтов портов не выполняется. Если    #
#               указанный порт уже присвоен другой службе сервера, #
#               демон возвратит сообщение об ошибке и не будет    #
#               запущен. Проверьте протокол SYSLOG и убедитесь,   #
#               что все серверы запущены успешно.                 #
#                                                                 #
#                                                                 #
# [Пример]                                                        #
#  Authentication_Ports  : 1812,6666 (без пробелов между запятыми) #
#                                                                 #
#   В примере, приведенном выше, для каждого указанного порта     #
#   будет запущен отдельный сервер. В данном случае               #
#                                                                 #
#           6666 : порт 6666                                      #
#                                                                 #
#-----------------------------------------------------------------#
Authentication_Ports  : 1812
Accounting_Ports      : 1813
#-----------------------------------------------------------------#
#       Информация о пользователях из каталога LDAP               #
#                                                                 #
#   Данные атрибуты необходимы, если RADIUS взаимодействует с     #
#   каталогом LDAP версии 3 и в поле Database_location указано    #
#   значение LDAP                                                 #
#                                                                 #
#   LDAP_User     : ИД пользователя, обладающего правами доступа к #
#                   удаленной базе данных (LDAP). В данном случае #
#                   это отличительное имя администратора LDAP.    #
#                                                                 #
#   LDAP_User_Pwd : Пароль доступа к каталогу LDAP, связанный с   #
#                   приведенным выше ИД пользователя.             #
```

```
#                                                                     #
#---------------------------------------------------------------------#
LDAP_User        : cn=root
LDAP_User_Pwd    :
#---------------------------------------------------------------------#
#       Сведения о каталоге LDAP                                      #
#                                                                     #
#   Если в поле Database_location указано значение "LDAP",            #
#   следующие поля должны быть заполнены.                             #
#                                                                     #
#   LDAP_Server_name     : Полное имя хоста сервера LDAP версии 3.    #
#   LDAP_Server_Port     : Номер порта TCP, применяемого сервером     #
#                          LDAP. По умолчанию для LDAP применяется    #
#                          порт 389.                                  #
#   LDP_Base_DN          : Отличительное имя для запуска поиска       #
#   LDAP_Timeout         : # секунд ожидания ответа от                #
#                          сервера LDAP                               #
#   LDAP_Hoplimit        : Максимальная длина последовательности      #
#                          переадресации                              #
#   LDAP_Sizelimit       : Ограничение размера для поиска             #
#                          (число записей)                            #
#   LDAP_Debug_level     : 0=Трассировка выключена, 1=Трассировка     #
#                          включена                                   #
#                                                                     #
#---------------------------------------------------------------------#
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
#---------------------------------------------------------------------#
#       Информация о PROXY RADIUS                                     #
#                                                                     #
#                                                                     #
#   Proxy_Allow               :  ON или OFF. Если значение равно ON,#
#                                сервер может передать пакеты в из-#
#                                вестные ему области. Следует на-  #
#                                строить также следующие ключи.   #
#   Proxy_Use_Table           :  ON или OFF. Если значение равно ON,#
#                                то для ускорения обработки или дуб-#
#                                лирования запросов может применять-#
#                                ся таблица. Может изменяться без   #
#                                Proxy ON, но если Proxy_Use_Table = ON,#
#                                то его значение должно равняться ON.     #
#   Proxy_Realm_name          :  Задает область, обслуживаемую     #
#                                этим сервером.                     #
#   Proxy_Prefix_delim        :  Список разделителей имен          #
#                                областей, указанных в качестве    #
#                                префикса имени пользователя. В    #
#                                этом списке недопустимы символы   #
#                                из списка разделителей суффикса.  #
#   Proxy_Suffix_delim        :  Список разделителей имен          #
#                                областей, указанных в качестве    #
#                                суффикса имени пользователя. В    #
#                                этом списке недопустимы символы   #
#                                из списка разделителей префикса.  #
#   Proxy_Remove_Hops         :  YES или NO. Если указано значение #
#                                YES, удаляется имя текущей        #
#                                области, имена предыдущих         #
#                                транзитных областей, а также имя  #
#                                области следующего сервера.       #
#                                                                  #
#   Proxy_Retry_count         :  Число попыток отправки пакета с   #
#                                запросом.                          #
#                                                                  #
```

```
#   Proxy_Time_Out            :  Число секунд ожидания             #
#                                   между попытками передачи.      #
#                                                                  #
#------------------------------------------------------------------#
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
#------------------------------------------------------------------#
#    Конфигурация идентификации локальной операционной системы     #
#                                                                  #
#  UNIX_Check_Login_Restrictions  : ON или OFF. Если ON, то при    #
#                                   идентификации локальной ОС     #
#                                   для проверки наличия локальных #
#                                   ограничений пользователя на    #
#                                   вход будет вызвана функция     #
#                                   loginrestrictions().   #
#                                                                  #
#------------------------------------------------------------------#
UNIX_Check_Login_Restrictions : OFF
#------------------------------------------------------------------#
#    Флаг глобального пула IP                                      #
#                                                                  #
#  Enable_IP_Pool : ON или OFF. Если ON, то сервер RADIUS будет    #
#                   присваивать IP-адрес из пула адресов,          #
#                   определенного для сервера RADIUS.                 #
#                                                                  #
#------------------------------------------------------------------#
Enable_IP_Pool            :   OFF
#------------------------------------------------------------------#
# Send Accept MA: ON или OFF. Некоторые службы NAS работают        #
#     некорректно, если идентификаторы сообщений (MA) присутствуют #
#     в сообщении ACCEPT. Этот параметр используется для           #
# выключения MA при отправке сообщения ACCEPT.                 #
#                           #
#                                                                  #
# ПРИМЕЧАНИЕ: иногда службы NAS также несовместимы с               #
#             нестандартными сообщениями ACCEPT.                   #
#                                                #
#                                                                  #
#------------------------------------------------------------------#
Send_Accept_MA : ON
#------------------------------------------------------------------#
#                                                                  #
# Maximum_Threads : число потоков, порождаемых                     #
#                   для обработки запросов на                      #
#                   идентификацию. Если значение не указано,       #
#                   RADIUS по умолчанию использует 10.                    #
#                                                                  #
#------------------------------------------------------------------#
Maximum_Threads : 99
#------------------------------------------------------------------#
#                                                                  #
# EAP_Conversation_Timeout : время ожидания в секундах             #
#                            перед тем, как диалог                 #
#                            становится неактуальным и удаляется.          #
#                                                                  #
# ПРИМЕЧАНИЕ: этот параметр предотвращает атаки типа               #
#   "Отказ в обслуживании" (DoS) на RADIUS Authentication Server.  #
# Если в сети наблюдаются большие задержки, возможно,          #
#   значение этого параметра
потребуется увеличить.                                               #
#                                                                  #
```

```
#-------------------------------------------------------------------#
EAP_Conversation_Timeout : 30
#-------------------------------------------------------------------#
# Глобальные параметры EAP-TLS (eap-tls):                           #
#                                                                   #
# Примеры:                                                          #
#                                                                   #
# Enable_EAP-TLS : ON или OFF. Если значение равно ON, сервер       #
#       может использовать OpenSSL для идентификации пользователей #
#       с помощью EAP-TLS. Эти пользователи должны иметь тип        #
#       идентификации EAP 13 (или EAP-TLS).                         #
# Значение это параметра можно узнать с помощью smitty:      #
#       'smitty rad_conf_users'                             #
#                                                                   #
# ПРИМЕЧАНИЕ: следующие атрибуты полностью игнорируются,            #
#       когда атрибут "Enable_EAP" имеет значение "OFF".            #
#                                                                   #
# 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                     #
#                                                                   #
# ПРИМЕЧАНИЕ: Server_Cert_File и Server_PrivKey_File могут быть     #
#       одним и тем же файлом, если файл имеет следующий формат     #
#       (в любом порядке):                                          #
#                                                                   #
#       -----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     :
```

Методы идентификации EAP для каждого пользователя можно настроить с помощью программы SMIT. Для настройки методов EAP для каждого пользователя, выполните следующие действия:

```
Radius Server
     -> Configure users
          -> Local Database
             LDAP Directory
                  -> Add a user
                     Change/Show Characteristics of a user
                          ->
                          Login User ID [ ]
                          EAP Type [0 2 4]
                          Password Max Age
```

При выборе типа EAP доступны следующие варианты:

0 Нет

2 MD5 - challenge

4 TLS

Для идентификации производится сравнение выбранного метода EAP с последовательностью метода идентификации, заданной в файле `radiusd.conf`.

Файл /etc/radius/clients:

В файле `clients` перечислены клиенты, которым разрешено отправлять запросы серверу RADIUS.

Как правило, для каждого клиента (NAS или AP) вводится IP-адрес клиента, общий шифр для сервера RADIUS и клиента, а также необязательное *имя-пула* для пула IP.

Для записей в этом файле применяется следующий формат:

```
<Client IP Address>    <Shared Secret>    <Pool Name>
```

Пример списка записей:

```
10.10.10.1      mysecret1      floor6
10.10.10.2      mysecret2      floor5
```

Общий шифр - это символьная строка, настроенная на клиентском оборудовании и на сервере RADIUS. Максимальная длина общего шифра равна 256 байтам, регистр букв учитывается. Общий шифр не передается в пакетах RADIUS и никогда не передается по сети. Администратор должен убедиться, что общий шифр на обеих сторонах (клиент и сервер RADIUS) точно совпадает. Общий шифр предназначен для шифрования пароля пользователя и может применяться для проверки целостности сообщений (посредством атрибута Message Authentication).

Общий шифр каждого клиента должен быть уникальным в файле`/etc/radius/clients` и, как хороший пароль, рекомендуется составлять его из букв, цифр и символов верхнего и нижнего регистров. Для большей гарантии защиты рекомендуется, чтобы шифр был длиной не менее 16 символов. Файл `/etc/radius/clients` можно изменить с помощью программы SMIT. Во избежание атак по словарю шифр рекомендуется часто менять.

Имя-пула - это имя пула, из которого при динамической трансляции адресов берутся глобальные IP-адреса. *Имя-пула* создается системным администратором при настройке сервера RADIUS. С помощью панели SMIT *имя-пула* добавляется так: Меню **Настроить правила Proxy > Пул IP-адресов > Создать пул IP-адресов**. Используется при выделении пула IP-адресов на сервере.

Файл /etc/radius/dictionary:

Файл `dictionary` содержит описания атрибутов, определенных протоколом RADIUS и поддерживаемых сервером AIX RADIUS.

Он применяется демоном RADIUS для создания и проверки данных пакетов. В данный файл следует добавить атрибуты, определенные вендорами. Файл dictionary можно изменить с помощью любого редактора. Отдельный интерфейс SMIT для этого не предусмотрен.

Ниже приведен фрагмент примера файла dictionary:

```
#########################################################################
#                                                                       #
#   В этом файле приведены записи словаря, предназначенные для анализа #
#   запросов и создания ответов. Все записи представляют собой пары    #
#   атрибут/значение. Для атрибутов допустимы значения следующих       #
#   четырех типов:                                                      #
#                                                                       #
#   string - 0-253 октетов                                              #
#   ipaddr - 4 октета в десятичном формате с точками                    #
#   integer - 32-разрядное значение со старшим байтом в начале          #
```

```
#   date - 32-разрядное значение со старшим байтом вначале - число     #
#          секунд, прошедших с 00:00:00 GMT,  1 января 1970 года       #
#                                                                       #
#   Для простоты администрирования перечисленные значения хранятся в   #
#   файле пользователя вместе с записями словаря VALUE.                 #
#                                                                       #
#   Пример:                                                             #
#                                                                       #
#   ATTRIBUTE         VALUE                                             #
#   ---------------   -----                                             #
#   Framed-Protocol = PPP                                               #
#   7               = 1     (кодирование на основе целых)               #
#                                                                       #
#########################################################################
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
.
.
.
```

Примечание: В файлах `default.policy` и `default.auth` (либо пользовательских файлах `user_id.policy` и `user_id.auth`) можно указывать только допустимые атрибуты RADIUS, описанные в локальном файле конфигурации словаря AIX. Если атрибут не указан в словаре, то в протокол заносится сообщение об ошибке и загрузка демона **radiusd** не выполняется.

Примечание: Для применения изменений, внесенных в файлы `default.policy` и `default.auth` следует перезапустить демонов RADIUS с помощью команд **stopsrc** и **startsrc**, либо с помощью программы SMIT.

Файл /etc/radius/proxy:

Файл `/etc/radius/proxy` - это файл конфигурации, применяемый службами Proxy. Этот файл применяется для преобразования известных областей, в которые сервер Proxy может пересылать пакеты.

Файл `/etc/radius/proxy` содержит IP-адреса серверов, обрабатывающих пакеты в известных областях, а также общие шифры для взаимодействия с этими серверами.

Файл содержит следующие поля, которые можно изменять с помощью программы SMIT:

- **Имя области**
- **IP адрес следующего узла**
- **Общий ключ**

Ниже приведен пример файла `/etc/radius/proxy`:

Примечание:

Длина общего шифра может быть до 16 символов. Для следующего транзитного сервера RADIUS должен быть указан такой же общий шифр.

```
# @(#)91 1.3  src/rad/usr/sbin/config_files/proxy, radconfig, radius530 1/23/04 13:11:14
#########################################################################
#                                                                       #
#       В этом файле перечислены области Proxy, обладающие правами на  #
```

```
#       обмен пакетами с запросами и ответам Proxy с сервером, а      #
#       также применяемые общие шифры.                                 #
#                                                                      #
#       Первый столбец - это имя области удаленного сервера RADIUS.   #
#                                                                      #
#       Второй столбец - это IP-адрес удаленного сервера RADIUS в      #
#       допустимом формате.                                            #
#                                                                      #
#       Третий столбец - это общий шифр, связанный с областью.         #
#                                                                      #
#       Примечание: В этом файле содержится конфиденциальная           #
#                   информация, для защиты которой следует             #
#                   предпринять соответствующие меры.                  #
#                                                                      #
########################################################################
# REALM NAME               REALM IP               SHARED SECRET
#----------------       -------------------     ----------------------
#  myRealm                10.10.10.10              sharedsec
```

Идентификация

Стандартный способ идентификации предусматривает ввод имени и фиксированного пароля при входе пользователя в систему или отправке запроса на службу. ИД пользователей, пароли и прочая информация, необходимая для работы сервера RADIUS, хранится в базе данных идентификации.

Для идентификации пользователей сервер поддерживает локальную базу данных, пароли UNIX и LDAP. Расположение базы данных задается в файле `/etc/radius/radiusd.conf` в процессе настройки. При необходимости вы можете изменить его с помощью программы SMIT. Дополнительная информация о файлах конфигурации RADIUS приведена в разделе “Файлы конфигурации RADIUS” на стр. 320.

Базы данных пользователей:

Программное обеспечение RADIUS позволяет хранить информацию пользователей в различных базах данных.

Для хранения информации пользователей можно использовать локальную базу данных, базу данных UNIX или LDAP.

UNIX:

Опция идентификации UNIX позволяет серверу RADIUS использовать для идентификации пользователей метод идентификации локальной системы.

Для применения локальной идентификации UNIX измените поле **database_location** в файле`radiusd.conf` или выберите значение `UNIX` в поле `Расположение базы данных` программы SMIT. Данный способ идентификации предусматривает проверку ИД пользователя и пароля с помощью интерфейса прикладных программ **authenticate()** UNIX. Пароли сохраняются в том же файле, который используется в UNIX, например, `/etc/passwords`. ИД пользователей и пароли в этом случае создаются с помощью команды **mkuser** или SMIT.

Для того чтобы использовалась база данных UNIX, выберите значение `UNIX` в поле **Расположение базы данных**, как это показано ниже:

```
    Настроить сервер

Каталог RADIUS                                      /etc/radius
*Расположение базы данных                           [UNIX]
Имя файла локальной базы данных AVL                 [dbdata.bin]
Локальный учет                                      [Вкл]

Уровень отладки                                     [3]
.
.
.
```

Локальная база данных:

Если в поле **database_location** файла `radiusd.conf` или записи Расположение базы данных программы SMIT указано значение`Local`, то все ИД пользователей и пароли сервер RADIUS будет хранить в файле `/etc/radius/dbdata.bin`.

Локальная база данных пользователей представляет собой простой файл, содержащий сведения об ИД пользователей и паролях. Пароли в этом файле защищены с помощью хэширования. Хэширование - это быстрый способ прямого обращения к данным в области памяти. Добавить, удалить и изменить пароли пользователей можно с помощью команды **raddbm** или SMIT. При запуске демон **radiusd** считывает файл `radiusd.conf` и загружает ИД пользователей и пароли в память.

Примечание: Максимальная длина ИД пользователя составляет 253 символа, максимальная длина пароля - 128 символов.

Для применения локальной базы данных выберите значение `Локальное` в поле **Расположение базы данных**, как это показано ниже:

```
    Настроить сервер

Каталог RADIUS                                      /etc/radius
*Расположение базы данных           [Локальное]
Имя файла локальной базы данных AVL                 [dbdata.bin]
Локальный учет                                      [Вкл]

Уровень отладки                                     [3]
.
.
.
```

LDAP:

RADIUS поддерживает LDAP версии 3 для хранения сведений об удаленных пользователях.

Для обращения к этим данным в RADIUS применяются удаленные вызовы API LDAP версии 3. Для работы с LDAP версии 3 в поле **database_location** файла `/etc/radiusd.conf` должно быть указано значение `LDAP`. Кроме того, должны быть заданы имя сервера, ИД пользователя и пароль администратора LDAP.

Система AIX работает с библиотеками клиента LDAP версии 3, которые поставляются вместе с сервером каталогов IBM Tivoli. LDAP представляет собой масштабируемый протокол, одним из преимуществ которого является простота администрирования сервера RADIUS, обусловленная централизованным расположением сведений о пользователях и обрабатываемых данных. Данные RADIUS можно просмотреть, выполнив с помощью командной строки команду **ldapsearch**.

Сервер LDAP необходимо настроить перед применением совместно с RADIUS.

В состав программного обеспечения RADIUS входят файлы `ldif`, предназначенные для добавления в каталог LDAP схемы RADIUS, в частности классов объектов и атрибутов. Однако предварительно необходимо установить и настроить сервер LDAP.

Для объектов RADIUS LDAP создается отдельный суффикс. Данный суффикс представляет собой контейнер с именем cn=aixradius, в котором расположены два класса объектов, описанные “Конфигурация сервера LDAP RADIUS”. Суффикс и схема RADIUS создаются с помощью файла `ldif`, который поставляется вместе с RADIUS.

Ниже перечислены возможности, которые становятся доступными в результате применения LDAP в качестве базы данных идентификации:

1. База данных, доступная всем серверам RADIUS.
2. Список активных пользователей
3. Ограничение числа входов в систему для отдельных ИД пользователей
4. Тип **EAP**, который можно настроить для одного пользователя
5. Срок действия пароля.

Для применения базы данных LDAP выберите значение `LDAP` в поле **Расположение базы данных**, как это показано ниже:

```
    Настроить сервер

Каталог RADIUS                                  /etc/radius
*Расположение базы данных           [LDAP]
Имя файла локальной базы данных AVL             [dbdata.bin]
Локальный учет             [ON]

Уровень отладки                                 [3]
.
.
.
```

Информация, связанная с данной:

IBM Directory Server

Конфигурация сервера LDAP RADIUS:

После настройки идентификации пользователей LDAP следует обновить схему сервера LDAP. Администратор сервера LDAP должен добавить соответствующие атрибуты AIX RADIUS, а также классы объектов до того, как определять пользователей LDAP RADIUS.

Для сервера LDAP необходимо указать суффикс. Для RADIUS применяется суффикс `cn=aixradius`. Суффикс - это отличительное имя, указывающее на корневую запись в структуре каталогов.

При добавлении суффикса на сервере LDAP создается пустой контейнер. *Контейнер* - это пустая запись, применяемая для разделения пространства имен. Контейнер, в котором можно создавать дочерние записи, аналогичен каталогу файловой системы. Информация из пользовательского профайла добавляется в каталог LDAP с помощью программы SMIT. ИД пользователя и пароль администратора LDAP хранятся в файле `/etc/radius/radiusd.conf`. При необходимости вы можете изменить их с помощью программы SMIT сервера RADIUS.

Для организации информации, хранящейся в каталоге LDAP, в схеме определяются классы объектов. Класс объектов состоит из набора обязательных и необязательных атрибутов. Атрибуты представлены в виде пар `тип=значение`, в которых тип определяется уникальным идентификатором объекта (OID), а значение должно соответствовать заданному формату. Каждая запись каталога LDAP представляет собой экземпляр объекта.

Примечание: Класс объектов сам по себе не описывает структуру каталогов или пространство имен. Для этого требуется дополнительно создать записи и присвоить конкретным экземплярам классов объектов уникальные отличительные имена. Например, классу объектов контейнера присвоено уникальное отличительное имя, его можно связать с двумя другими записями, которые представляют собой экземпляры структурных единиц класса объектов. В результате создается древовидная структура или пространство имен.

Классы объектов могут быть связаны не только с сервером RADIUS. Они загружаются из файла `ldif`. Некоторые атрибуты - это атрибуты существующей схемы LDAP, другие атрибуты характеры только для RADIUS. Новые классы объектов RADIUS являются структурными и абстрактными.

По причинам, связанным с защитой, для связывания с сервером LDAP применяется простая связь или вызов API SASL **ldap_bind_s**, который включает в себя DN, способ идентификации CRAM-MD5, а также пароль администратора LDAP. Такой подход предусматривает передачу по сети описателей сообщений вместо паролей. CRAM-MD5 - это механизм защиты, для работы которого не требуется дополнительная настройка взаимодействующих сторон (клиента или сервера).

Примечание: Для всех атрибутов классов объектов допустимо только одно значение.

Пространство имен LDAP RADIUS:

На верхнем уровне структуры пространства имен RADIUS LDAP расположен контейнер `cn=aixradius`. На следующем уровне под `cn=aixradius` расположены две структурные единицы (OU). OU - это контейнер, позволяющий сделать записи уникальными.

На следующем рисунке представлена схема LDAP RADIUS. На этом рисунке показаны контейнеры и структурные единицы, представленные в виде окружностей, соединенных линиями (ветвями). Контейнер `aixradius`, расположенный в центре, разветвляется на две структурные единицы: `ibm-radiususer` и `ibm-radiusactiveusers`. Контейнер `ibm-radiususer` содержит дочерние контейнеры `userid`, `password` и `maxLogin`. Контейнер `ibmradiusactiveusers` содержит контейнеры `userid +`, `login number`, `login status` и `session_id`. Над контейнером `aixradius` расположены контейнеры `aixsecurity` и `root`.

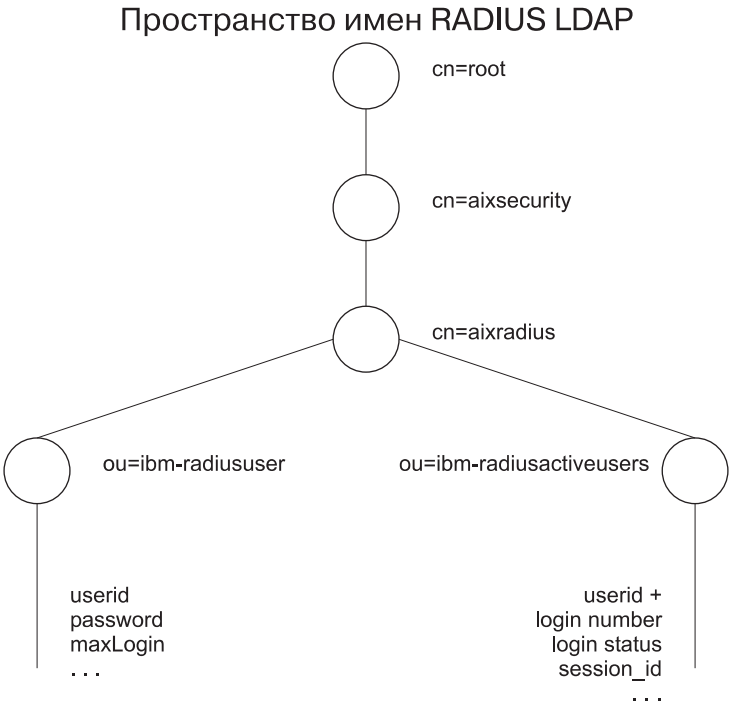


Рисунок 16. Пространство имен LDAP RADIUS

Файлы схемы пространства имен LDAP:

Файлы схемы LDAP определяют классы объектов и особые атрибуты RADIUS для пространства имен LDAP.

Следующие файлы схемы LDAP находятся в каталоге `/etc/radius/ldap`:

`IBM.V3.radiusbase.schema.ldif`

Данный файл определяет класс объектов верхнего уровня для сервера RADIUS (`cn=aixradius`). Кроме того, в этом файле указаны следующие ветви класса объектов `cn=aixradius`:

```
ou=ibm-radiususer
ou=ibm-radiusactiveusers
```

Для добавления необходимой информации воспользуйтесь следующей командой:

```
ldapadd -D ИД-администратора-ldap -w пароль -i /etc/radius/ldap/IBM.V3.radiusbase.schema.ldif
```

Эту команду можно выполнить как на сервере LDAP, так и в удаленной системе, с помощью опции **-h** (имя хоста).

`IBM.V3.radius.schema.ldif`

Данный файл определяет особые атрибуты RADIUS и классы объектов.

Следующая команда позволяет добавить новые атрибуты и классы объектов RADIUS:

```
ldapmodify -D ИД-администратора-ldap -w пароль -i /etc/radius/ldap/IBM.V3.radius.schema.ldif
```

Кроме того, с помощью программы SMIT необходимо указать `LDAP` в качестве расположения базы данных, а также задать имя сервера LDAP и пароль администратора. После этого можно добавить пользователей RADIUS LDAP в каталог с помощью программы SMIT.

Класс объектов пользовательского профайла:

Пользовательские профайлы LDAP необходимо добавить в систему до применения сервера RADIUS в целях идентификации. Профайл содержит ИД пользователя и пароль.

Объекты пользовательских профайлов содержат сведения о конкретных сотрудниках, обладающих доступом к сети, а также идентификационную информацию. Обращение к классу объектов **ibm-radiusUserInstance** выполняется одновременно с вызовом API LDAP из демона. Уникальное поле, с которого начинается отличительное имя, представляет собой ИД пользователя. Поле **MaxLoginCount** позволяет ограничить число входов пользователя LDAP в систему.

Класс объектов списка зарегистрированных пользователей:

В списке зарегистрированных пользователей LDAP представлены сведения о пользователях, которые в настоящий момент работают в системе.

Для каждого пользователя предусмотрено несколько записей, начиная с `login_number = 1` и заканчивая `MaxLoginCount = 5`. ИД сеанса извлекается из сообщения RADIUS `start_accounting`. При создании объекта **ibm-radiusUserInstance** создаются частично завершенные записи. Это означает, что большинство полей этих записей заполняются только после получения пакетов учета RADIUS. В результате получения сообщения RADIUS `start_accounting` в объект **ibm-radiusactiveusers** заносятся сведения о том, что пользователь зарегистрирован в системе, а для соответствующего регистрационного номера записываются уникальные сведения о сеансе. После получения сообщения `stop_accounting` информация из записи списка зарегистрированных пользователей удаляется. В запись списка зарегистрированных пользователей заносятся сведения о том, что пользователь вышел из системы. В сообщениях запуска и завершения учета указаны одинаковые уникальные номера сеансов. Одновременно с помощью вызовов API LDAP выполняется обращение к соответствующему классу объектов.

Протокол идентификации по паролю:

Защита, обеспечиваемая протоколом идентификации по паролю (**PAP**), основана на кодировании паролей пользователей с помощью алгоритма с хэшированием MD5.

Ниже описано, как это происходит:

1. В пакетах с паролем пользователя поле идентификации, содержит случайное число, длина которого составляет шестнадцать октетов, называемое Идентификатором запроса.
2. Идентификатор запроса и общий шифр клиента преобразуются в хэш-код MD5. Результатом является хэш длиной 16 октетов.
3. Пароль, указанный пользователям, дополняется нулями до шестнадцати октетов.
4. Хэш-код, полученный на шаге 2, объединяется с паролем с помощью логической операции XOR (исключающее ИЛИ). Полученный результат отправляется в пакете в качестве атрибута *user_password*.
5. Сервер RADIUS рассчитывает хэш-код, указанный в шаге 2.
6. Пароль восстанавливается путем объединения данных из пакета (см. шаг 4) с хэш-кодом с помощью операции XOR.

Протокол идентификации по квитированию вызова:

Кроме того, RADIUS поддерживает защиту паролей с помощью протокола PPP **CHAP**.

Протокол CHAP не предусматривает передачу паролей пользователей по сети. Вместо этого отправляется хэш-код MD5 пароля, который затем восстанавливается на сервере RADIUS и сравнивается с паролем, указанным в информации о пользователе.

Расширяемый протокол идентификации:

Расширяемый протокол идентификации (**EAP**) - это протокол, позволяющий обеспечить поддержку нескольких способов идентификации.

EAP задает структуру взаимодействия между клиентом и сервером идентификации без четкого определения передаваемых при этом идентификационных данных. Идентификационные данные в свою очередь задаются конкретным способом идентификации **EAP**. **EAP** поддерживает следующие способы идентификации:

- MD5-challenge
- One-time password
- Generic token card
- TLS

Реализация протокола **EAP** в RADIUS предусматривает указание атрибутов RADIUS, применяемых для передачи данных **EAP** между серверами RADIUS и клиентами. Сервер RADIUS передает полученные данные **EAP** базовым серверам, в которых реализованы различные способы идентификации **EAP**.

Сервер RADIUS AIX поддерживает только идентификацию EAP-TLS и MD5-challenge EAP.

Способ идентификации EAP задается на уровне пользователя. Для этого в записи пользователя, расположенной в локальной базе данных или каталоге LDAP, указывается соответствующее значение.

По умолчанию протокол EAP выключен для всех пользователей.

Права доступа

RADIUS позволяет указать отдельные атрибуты прав доступа для каждого пользователя в файлах стратегии прав доступа `default.auth` и `default.policy`.

Атрибуты прав доступа - это допустимые атрибуты протокола RADIUS, описанные в соответствующем RFC и определенные в файле `/etc/radius/dictionary`. Предоставление прав доступа выполняется не всегда в зависимости от конфигурации аппаратного обеспечения NAS или точки доступа. При необходимости настройте атрибуты прав доступа. Предоставление прав доступа выполняется только после успешной идентификации.

Стратегия - это настраиваемый набор пар атрибут-значение, управляющий работой пользователей в сети. Стратегии могут применяться как на глобальном уровне для всего сервера RADIUS, так и для конкретных пользователей.

В комплект поставки входят два файла конфигурации предоставления прав доступа: `/etc/radius/authorization/default.auth` и `default.policy`. Файл `default.policy` применяется для обработки входящих пакетов с запросами на предоставление доступа. Изначально в этом файле содержатся пустые пары атрибут-значение, которые следует дополнительно настроить. После идентификации в соответствии со стратегией клиенту возвращается пакет разрешения доступа или запрета доступа.

Кроме того, для каждого пользователя можно создать файл *ИД-пользователя*`.policy`. Атрибуты, указанные в уникальном файле стратегии, связанном с ИД пользователя, проверяются в первую очередь. Если соответствующий набор пар атрибут-значение в файле *ИД-пользователя*`.policy` не найден, проверяется файл `default.policy`. Если пары атрибутов из пакета с запросом на предоставление доступа не найдены ни в одном из файлов, то клиенту возвращается пакет запрета доступа. Если совпадение найдено в одном из файлов, возвращается пакет разрешения доступа. Такой подход позволяет эффективно реализовать два уровня стратегии.

Файл `default.auth` содержит список пар атрибут-значение, возвращаемых клиенту после проверки стратегии. Изначально в файле `default.auth` также содержатся пустые пары атрибут-значение, которые следует дополнительно настроить. Вы можете непосредственно внести изменения в файл `default.auth`, либо настроить необходимые параметры предоставления прав доступа с помощью SMIT. Все атрибуты, для которых указаны значения, автоматически возвращаются серверу сетевого доступа (NAS) в пакете разрешения доступа.

Кроме того, атрибуты прав доступа для отдельного пользователя можно указать в файле с именем, состоящим из уникального ИД пользователя и расширения .auth, например: ***ИД-пользователя***`.auth`. Этот файл должен находится в каталоге `/etc/radius/authorization`. Для создания таких файлов и внесения в них изменений предусмотрена отдельная панель в программе SMIT.

Атрибуты прав доступа, указанные для конкретного пользователя, возвращаются в пакете разрешения доступа вместе с атрибутами по умолчанию из файла `default.auth`.

Если в файлах `default.auth` и *ИД-пользователя*`.auth` указаны одинаковые записи, то значения пользователя имеют больший приоритет, чем значения по умолчанию. Такой подход позволяет кроме глобальных атрибутов предоставления прав доступа (службы и ресурсы), которые присваиваются всем пользователя, применять более конкретные атрибуты для отдельных пользователей.

Примечание: Для объединения атрибутов идентификации со специфичными для пользователя атрибутами идентификации вместо файла `default.auth` следует использовать файл `global.auth`, если не требуется другой вид объединения.

Начиная с AIX версии 6.1 (технологический пакет обслуживания 6100-02) RADIUS поддерживает файл идентификации `global.auth`. Этот файл используется для объединения атрибутов идентификации, специфичных для пользователя, (в файлах `ИД-пользователя.auth`) с глобальными атрибутами идентификации.

В отличие от атрибутов файла `default.auth`, которые переопределяются атрибутами пользовательских файлов, атрибуты файла `global.auth` объединяются с ними, предоставляя большую гибкость в настройке.

Если атрибут присутствует и в файле `default.auth`, и в файле `ИД-пользователя.auth`; используется значение атрибута из файла пользователя. Такое переопределение значений по умолчанию позволяет часть атрибутов идентификации (служб, ресурсов) назначить всем пользователям, а остальные назначать для каждого пользователя индивидуально.

Это справедливо и для файла `global.auth` за исключением того, что он не переопределяется файлом `ИД-пользователя.auth`. Вместо переопределения файлы объединяются. Это удобно при использовании атрибутов, определяемых поставщиком.

Процесс предоставления прав доступа выполняется следующим образом:

1. В момент запуска демона списки идентификации и стратегии по умолчанию из файлов `/etc/radius/authorization/default.policy` и `default.auth` считываются в память.
2. Выполняется идентификация ИД пользователя и пароля.
3. Проверяются пары атрибут-значения, указанные во входящем пакете.
   a. Проверяется пользовательский файл ***ИД-пользователя*****`.auth`**.
   b. Если соответствие не найдено, проверяется файл `default.policy`.
   c. Если соответствие снова не найдено, то отправляется пакет запрета доступа.
4. Применяются пользовательские атрибуты прав доступа, если они указаны.
   a. Сравнивается содержимое файлов `/etc/radius/authorization/`***ИД-пользователя*****`.auth`** и `default.auth`.
   b. Записи из пользовательского файла переопределяют записи по умолчанию.
   c. Полученные в результате значения атрибутов объединяются с атрибутами файла `global.auth`.
5. Атрибуты предоставления прав доступа возвращаются в пакете разрешения доступа.

Учет

Сервер учета RADIUS принимает запросы, связанные с учетом, от клиентов и возвращает им подтверждения получения и сохранения учетных данных.

В файле `radiusd.conf` можно включить локальный учет.

Если учет RADIUS включен в конфигурации клиента, создается пакет ACCOUNTING_START, содержащий сведения о типе предоставляемой службы, пользователе, которому она предоставляется, а также начальном времени предоставления. Клиент отправляет этот пакет серверу учета RADIUS, который в свою очередь возвращает уведомление о приеме этого пакета. После истечения срока действия службы клиент создает пакет ACCOUNTING_STOP, содержащий сведения о типе службы, а также дополнительную статистику, такую как затраченное время, входящие и исходящие октеты и номера входящих и исходящих пакетов. Получив пакет ACCOUNTING_STOP, сервер учета RADIUS возвращает клиенту уведомление о его приеме.

Пакеты ACCOUNTING_START и ACCOUNTING_STOP отправляются серверу учета RADIUS по сети. Рекомендуется настраивать конфигурацию клиента таким образом, чтобы пакеты ACCOUNTING_REQUEST отправлялись до тех пор, пока не будет получено соответствующее уведомление о приеме. Кроме того, вы можете настроить сервер Proxy, позволяющий пересылать запросы клиента альтернативным серверам учета в том случае, если основной сервер недоступен. Дополнительная информация о службах Proxy приведена в разделе “Службы Proxy” на стр. 337.

Учетные данные сохраняются в стандартном формате RADIUS (*атрибут=значение*) в файле `/etc/var/radius/data/accounting`. В пакет вместе с учетными данными записывается системное время. Если сервер учета RADIUS не сможет сохранить пакет учета, подтверждение **Accounting_Response** отправлено не будет, а в файл `syslog` будет занесено сообщение об ошибке.

Файл /var/radius/data/accounting:

В файл /var/radius/data/accounting записываются данные в соответствии с содержимым пакетов ACCOUNTING START и ACCOUNTING STOP.

В ходе установки создается пустой файл /var/radius/data/accounting. В этот файл записываются данные в соответствии с содержимым пакетов ACCOUNTING START и ACCOUNTING STOP.

Ниже приведен пример данных, которые сервер AIX RADIUS заносит в файл /var/radius/data/accounting. В зависимости от конфигурации вашей системы эти данные могут быть другими.

Примечание:

- Убедитесь, что в файловой системе /var достаточно свободного места для обработки всех учетных данных.
- Для анализа данных из этого файла можно использовать сценарии Perl других фирм. Примеры сценариев, создающих отчеты на основе учетных данных приведены на Web-сайте http://www.pgregg.com/projects/radiusreport
- Пакеты учетных данных также можно пересылать через сервер Proxy.

```
четверг 27 мая 2004 г. 14.43.19
 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

четверг 27 мая 2004 г. 14.45.19
 NAS-IP-Address = 10.10.10.1
 NAS-Port = 1   <-- пользователь rod установил физическое соединение с системой через порт #1
 NAS-Port-Type = Async
 User-Name = "rod"
 Acct-Status-Type = Stop
 Acct-Authentic = RADIUS
 Service-Type = Framed-User
 Acct-Session-Id = "0000000C"   <-- обратите внимание, что идентификаторы сеанса
совпадают. В этом случае запуски соответствуют завершениям
 Framed-Protocol = PPP
 Framed-IP-Address = 10.10.10.2   <-- IP-адрес пользователя rod
 Acct-Terminate-Cause = User-Request  <-- пользователь прервал сеанс
 Acct-Input-Octets = 4016
 Acct-Output-Octets = 142
 Acct-Input-Packets = 35
 Acct-Output-Packets = 7
 Acct-Session-Time = 120  <-- время в секундах
 Acct-Delay-Time = 0
 Timestamp = 1085687119  <-- продолжительность сеанса составляет 120 секунд (2 минуты)
```

Службы Proxy

Службы Proxy позволяют серверу RADIUS пересылать запросы, поступающие от сервера NAS, другим серверам RADIUS и возвращать ответные сообщения серверу NAS. Работа службы Proxy основана на понятии области.

Сервер RADIUS может одновременно выполнять роль сервера Proxy и базового сервера. Такой подход применим как для пакетов идентификации, так и для пакетов учета. По умолчанию служба Proxy выключена в файле radiusd.conf.

Области:

Область - это идентификатор, добавляемый до или после значения, обычно указанного в атрибуте User-Name, в соответствии с которым сервер RADIUS определяет целевой сервер для запуска процесса идентификации или учета.

Ниже приведен пример использовании областей в RADIUS:

Пользователь *Джо* работает в компании *XYZ*, офис которой расположен в Сакраменто. Для этого расположения применяется область *SAC*. Однако в настоящее время *Джо* находится в командировке в Нью Йорке. Область Нью Йорка - *NYC*. Когда *Джо* входит в систему, принадлежащую области *NYC*, в атрибуте User-Name передается значение *SAC/Joe*. В этом случае сервер RADIUS области *NYC*, пересылает данный пакет серверу, выполняющему функции идентификации и учета для пользователей области *SAC*.

Атрибут области User-Name:

Атрибут **User-Name** определяет способ маршрутизации пакетов идентификации или учета между областями. Этот атрибут задает порядок областей, через которые пакет должен пройти до конечного сервера идентификации или учета.

Для маршрутизации пакетов в атрибут **User-Name** добавляются имена областей. Решение о том, какие именно области, описывающие путь пакета, следует добавить в атрибут **User-Name**, принимает администратор, настраивающий конфигурацию службы RADIUS. Транзитные области можно указывать как до, так и после атрибута **User-Name**. Как правило, для разделения областей применяется косая черта (/) перед атрибутом **User-Name** и амперсанд (&) после него. Разделители указываются в файле `radiusd.conf`. Анализ атрибута **User-Name** выполняется слева направо.

Пример атрибута **User-Name**, в котором указан только префикс:

```
USA/TEXAS/AUSTIN/joe
```

Пример атрибута **User-Name**, в котором указан только суффикс:

```
joe@USA@TEXAS@AUSTIN
```

При необходимости вы можете одновременно указать как префикс, так и суффикс. Обратите внимание, что транзитные области, указанные в пакете, анализируются слева направо и области суффикса обрабатываются после областей префикса. Идентификация или сохранение учетных данных можно выполнять только в одном узле.

В следующем примере применение обоих способов позволяет получить такие же результаты, как и в двух предыдущих:

```
USA/joe@TEXAS@AUSTIN
```

Настройка служб Proxy:

Конфигурация служб Proxy RADIUS хранится в файле `proxy`, находящемся в каталоге `/etc/radius` directory.

Начальный файл по умолчанию `proxy` содержит фиктивные записи. В файле proxy предусмотрено три поля: **Realm Name**, **Next Hop IP address** и **Shared Secret**.

Для настройки правил Proxy выберите необходимый вариант:

```
    Настроить правила Proxy

Показать список всех служб Proxy
Добавить Proxy
Показать или изменить параметры Proxy
Удалить Proxy
```

Опция **Показать список всех служб Proxy** отображает содержимое файла /etc/radius/proxy в виде таблицы. Заголовки столбцов:

```
realm_name   next_hop_address  shared_secret
```

Опция **Добавить Proxy** отображает следующее меню. Информация, указанная в этом меню, добавляется в конец файла /etc/radius/proxy.

На каждом транзитном участке цепочки Proxy между двумя серверами RADIUS применяется общий шифр. Общий шифр хранится в файле /etc/radius/proxy_file. Для каждого транзитного участка цепочки общий шифр должен быть уникальным.

Дополнительная информация о создании общих шифров приведена в разделе “Файл /etc/radius/clients” на стр. 327.

Для добавления Proxy выберите поля, как показано ниже:

```
            Добавить Proxy
*Имя области                                                    []  (не более 64 символов)
*IP-адрес следующего транзитного узла (десятичный формат с точками)  [xx.xx.xx.xx]
*Общий шифр                            []  (строка от 6 до 256 символов)
```

Опция **Показать или изменить параметры Proxy** отображает список имен областей. В этом списке, отображаемом во всплывающем окне, необходимо выбрать имя области.

Опция **Удалить Proxy** отображает список имен областей. В этом списке, отображаемом во всплывающем окне, необходимо выбрать имя области. Перед удалением выбранной области отображается всплывающее окно подтверждения удаления.

Ниже приведен раздел с информацией о конфигурации Proxy из файла radiusd.conf:

```
#-------------------------------------------------------------------#
#       Информация о PROXY RADIUS                                   #
#                                                                   #
#                                                                   #
#   Proxy_Allow               :  ON или OFF. Если значение равно ON,#
#                                сервер может передать пакеты в из-#
#                                вестные ему области. Следует на-  #
#                                строить также следующие ключи.   #
#   Proxy_Use_Table           :  ON или OFF. Если значение равно ON,#
#                                то для ускорения обработки или дуб-#
#                                лирования запросов может применять-#
#                                ся таблица. Может изменяться без   #
#                                Proxy ON, но если Proxy_Use_Table = ON,#
#                                то его значение должно равняться ON.     #
#   Proxy_Realm_name          :  Задает область, обслуживаемую     #
#                                этим сервером.                     #
#   Proxy_Prefix_delim        :  Список разделителей имен          #
#                                областей, указанных в качестве    #
#                                префикса имени пользователя. В    #
#                                этом списке недопустимы символы   #
#                                из списка разделителей суффикса.  #
#   Proxy_Suffix_delim        :  Список разделителей имен          #
#                                областей, указанных в качестве    #
```

```
#                                суффикса имени пользователя. В    #
#                                этом списке недопустимы символы   #
#                                из списка разделителей префикса.  #
#   Proxy_Remove_Hops         :  YES или NO. Если указано значение #
#                                YES, удаляется имя текущей        #
#                                области, имена предыдущих         #
#                                транзитных областей, а также имя  #
#                                области следующего сервера.       #
#                                                                  #
#   Proxy_Retry_count         :  Число попыток отправки пакета с   #
#                                запросом.                         #
#                                                                  #
#   Proxy_Time_Out            :  Число секунд ожидания             #
#                                между попытками передачи.         #
#                                                                  #
#------------------------------------------------------------------#
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
```

Настройка сервера RADIUS:

Демон сервера RADIUS использует несколько файлов конфигурации. Информация о конфигурации сервера сохраняется в файле `/etc/radius/radiusd.conf`. В поставляемом файле конфигурации указаны значения по умолчанию.

Примечание: Ниже приведен пример панели Настроить сервер RADIUS, предусмотренной в программе SMIT:

```
    Настроить сервер

Каталог RADIUS                                          /etc/radius
*Расположение базы данных                               [UNIX]
Имя файла локальной базы данных AVL                     [dbdata.bin]
Локальный учет                                          [Вкл]
Каталог локального учета                                []

Уровень отладки                                         [3]
Ответные сообщения о разрешении доступа                 []
Ответные сообщения о запрете доступа                    []
Ответные сообщения о вызове                             []
Ответные сообщения об истечении срока действия пароля   []
Поддержка продления просроченных паролей                [Нет]
Требовать идентификатор сообщения                       [Нет]

*Номер порта идентификации                              [1812]
*Номер порта учета                                      [1813]

Имя сервера LDAP                                        []
Номер порта сервера LDAP                                [389]
Отличительное имя администратора сервера LDAP           []
Пароль администратора сервера LDAP                      []
Базовое отличительное имя LDAP                          [cn=aixradius]
Ограничение размера LDAP                                [0]
Ограничение на пересылку LDAP                           [0]
Ограничение времени ожидания LDAP                       [10]
Уровень отладки LDAP                                    [0]

Применение Proxy                                        [Выкл]
Таблица использования Proxy                             [Выкл]
Имя области Proxy                                       []
Разделители префикса Proxy                              [$/]
Разделители суффикса Proxy                              [@.]
     Примечание: префикс и суффикс являются
                 взаимоисключающими
Удалять транзитные области Proxy                        [Нет]
Число попыток Proxy                                     [2]
Тайм-аут Proxy                                          [30]
Проверять ограничения на вход в систему UNIX            [Выкл]
Поддержка пула IP                                       [Вкл]
Последовательность метода идентификации                 [TLS, MD5]
Файл конфигурации OpenSSL                               []
```

Утилиты для ведения протоколов

Для занесения в протокол сведений об операциях и ошибках сервер RADIUS применяет SYSLOG.

В протокол заносится информация трех типов:

0 В протокол заносятся только неполадки, ошибки и сообщения о запуске демонов.

3 Ведение контрольного журнала сообщений `access_accept`, `access_reject*`, `discard` и `error`.

Примечание: Сообщения `discard` заносятся в протокол только в том случае, если при получении недопустимого входящего пакета не возвращается пакет ответа.

9 Информация, которая заносится в протокол на уровнях 0 и 3, а также дополнительная информация. Уровень 9 следует применять только в целях отладки.

По умолчанию применяется уровень ведения протокола 3. Такой подход позволяет повысить уровень контроля за сервером RADIUS. В зависимости от уровня ведения протокола сервера информацию, занесенную в протокол, можно проверить на подозрительные модели поведения. В случае нарушения защиты на основе вывода SYSLOG можно определить обстоятельства нарушения, а также права доступа, полученные в результате этого. Рекомендуется использовать данную информацию в процессе разработки дополнительных мер безопасности, позволяющих предотвратить возможные неполадки.

Информация, связанная с данной:

Настройка RADIUS для работы с демоном syslogd:

Для просмотра информации о задачах и ошибках с помощью SYSLOG необходимо включить демон syslogd.

Для настройки демона syslogd выполните следующие действия.

1. Добавьте в файл `/etc/syslog.conf` следующую строку `local4.debug var/adm/ipsec.log`. Для протоколирования событий защиты IP применяется функция *local4*. Используется стандартная система приоритетов операционной системы. Пока работа туннелей IP не стабилизируется, рекомендуется применять приоритет *debug*.

   Примечание: На ведение протокола могут потребоваться значительные вычислительные ресурсы и много дисковой памяти.
2. Сохраните файл `/etc/syslog.conf`.
3. Перейдите в каталог, в котором хранится файл протокола, и создайте пустой файл с таким же именем. В данном случае нужно перейти в каталог `/var/adm` и запустить следующую команду **touch**:

   ```
   touch ipsec.log
   ```
4. Запустите команду **refresh** для подсистемы `syslogd`:

   ```
   refresh -s syslogd
   ```

Настройка вывода SYSLOG:

Атрибут Debug_Level, расположенный в файле `radiusd.conf`, задает уровень отладки. Для него допустимы значения 0, 3 и 9, в зависимости от того, насколько подробная информация должна быть занесена в SYSLOG.

Значение по умолчанию - 3. Раздел отладки файла `radiusd.conf` выглядит приблизительно следующим образом:

```
#.
#.
#.
#   Debug_Level      :  Задает уровень отладки для сервера RADIUS. #
#                       Допустимые значения: 0, 3 и 9.             #
#                       По умолчанию применяется значение 3.       #
#                       Вывод отправляется в расположение,         #
#                       указанное в разделе *.debug в файле        #
#                       /etc/syslog.conf                           #
#                                                                  #
#                       С повышением уровня увеличивается число    #
#                       сообщений, отправляемых в файл syslog.     #
#                       Например, уровень "9" кроме собственных    #
#                       сообщений предусматривает отправку         #
#                       сообщений, создаваемых на уровнях          #
#                       "0" и "3".                                 #
#                                                                  #
#                       0 : минимальный уровень детализации        #
#                           протокола syslogd.  Для каждого         #
#                           процесса RADIUS заносятся сообщения о  #
#                           его начале и окончании.  Также заносятся#
#                           условия ошибок.                            #
#                                                                  #
#                       3 : заносятся сообщения ACCESS ACCEPT,     #
#                           REJECT и DISCARD для каждого пакета.   #
#                           Этот уровень обеспечивает ведение      #
#                           контрольного журнала идентификации.    #
#                                                                  #
#                       9 : Максимальная степень детализации. Ука- #
#                           зываются значения атрибутов при        #
```

```
#                          обработке транзакций, и так           #
#                          далее.                  #
#                      [Не рекомендуется в обычном режиме работы]  #
#                                                                  #
#------------------------------------------------------------------#
```

Ниже приведены примеры вывода различных уровней отладки.

Пакет учета с уровнем отладки 3

```
18 авг 10:23:57 server1 syslog: [0]:Процесс отслеживания [389288] запущен
18 авг 10:23:57 server1 radiusd[389288]: [0]:Локальная база данных (AVL) создана.
18 авг 10:23:57 server1 radiusd[389288]: [0]:Процесс идентификации запущен : Pid= 549082 Порт = 1812
18 авг 10:23:57 server1 radiusd[389288]: [0]:Процесс учета запущен : Pid= 643188 Порт = 1813
18 авг 10:23:57 server1 radiusd[643188]: [0]:Сокет создан [15]
18 авг 10:23:57 server1 radiusd[643188]: [0]:Ограничение сокета идентификации [15]
18 авг 10:23:57 server1 radiusd[549082]: [0]:Сокет создан [15]
18 авг 10:23:57 server1 radiusd[549082]: [0]:Ограничение сокета учета [15]
18 авг 10:24:07 server1 radiusd[643188]: [1]:*** Запуск Process_Packet() ***
18 авг 10:24:07 server1 radiusd[643188]: [1]:Код 4, ИД = 96, Порт = 41639 Хост = 10.10.10.10
18 авг 10:24:07 server1 radiusd[643188]: [1]:ACCOUNTING-START - отправка уведомления пользователю [ user_id1 ]
18 авг 10:24:07 server1 radiusd[643188]: [1]:Отправка уведомления с ИД 96 на адрес 10.10.10.10 (client1.ibm.com)
18 авг 10:24:07 server1 radiusd[643188]: [1]:send_acct_reply() Исходящий пакет:
18 авг 10:24:07 server1 radiusd[643188]: [1]: Код = 5, ИД = 96, Длина = 20
18 авг 10:24:07 server1 radiusd[643188]: [1]:*** Завершение Process_Packet() ***
18 авг 10:24:13 server1 radiusd[643188]: [2]:*** Запуск Process_Packet() ***
18 авг 10:24:13 server1 radiusd[643188]: [2]:Код 4, ИД = 97, Порт = 41639 Хост = 10.10.10.10
18 авг 10:24:13 server1 radiusd[643188]: [2]:ACCOUNTING-STOP - отправка уведомления пользователю [ user_id1 ]
18 авг 10:24:14 server1 radiusd[643188]: [2]:Отправка уведомления с ИД 97 на адрес 10.10.10.10 (client1.ibm.com)
18 авг 10:24:14 server1 radiusd[643188]: [2]:send_acct_reply() Исходящий пакет:
18 авг 10:24:14 server1 radiusd[643188]: [2]: Код = 5, ИД = 97, Длина = 20
18 авг 10:24:14 server1 radiusd[643188]: [2]:*** Завершение Process_Packet() **
```

Пакеты учета уровня 9

```
18 авг 10:21:18 server1 syslog: [0]:Процесс отслеживания [643170] запущен
18 авг 10:21:18 server1 radiusd[643170]: [0]:Локальная база данных (AVL) создана.
18 авг 10:21:18 server1 radiusd[643170]: [0]:Процесс идентификации запущен : Pid= 389284 Порт = 1812
18 авг 10:21:18 server1 radiusd[643170]: [0]:Процесс учета запущен : Pid= 549078 Порт = 1813
18 авг 10:22:03 server1 radiusd[643170]: [0]:PID = [389284] завершен
18 авг 10:22:03 server1 radiusd[643170]: [0]:PID = [549078] завершен
18 авг 10:22:03 server1 radiusd[643170]: [0]:Все дочерние процессы остановлены.
Завершение родительского процесса radiusd
18 авг 10:22:09 server1 syslog: [0]:Процесс отслеживания [1081472] запущен
18 авг 10:22:09 server1 radiusd[1081472]: [0]:Локальная база данных (AVL) создана.
18 авг 10:22:09 server1 radiusd[1081472]: [0]:Выполнение функции client_init()
18 авг 10:22:09 server1 radiusd[1081472]: [0]:Число прочитанных записей клиента: 1
18 авг 10:22:09 server1 radiusd[1081472]: [0]:Выполнение процедуры
read_authorize_policy s для файла /etc/radius/authorization/default.policy.
  /etc/radius/authorization/default.policy.
18 авг 10:22:09 server1 radiusd[1081472]: [0]:Внутри процедура read_authorize_file для файла:
  /etc/radius/authorization/default.policy.
18 авг 10:22:09 server1 radiusd[1081472]: [0]:процедура
read_authorize_file() завершена.
18 авг 10:22:09 server1 radiusd[1081472]: [0]:Внутри процедура read_authorize_file для файла:
  /etc/radius/authorization/default.auth.
18 авг 10:22:09 server1 radiusd[1081472]: [0]:процедура
read_authorize_file() завершена.
18 авг 10:22:09 server1 radiusd[549080]: [0]:connect_to_LDAP_server:Расположение базы данных (где находятся
  данные)=LDAP.
18 авг 10:22:09 server1 radiusd[549080]: [0]:connect_to_LDAP_server:
Имя сервера LDAP = server1.austin.ibm.com.
18 авг 10:22:09 server1 radiusd[549080]: [0]:connect_to_LDAP_server:
Порт сервера LDAP=389.
18 авг 10:22:09 server1 radiusd[1081472]: [0]:Процесс идентификации
запущен : Pid= 549080 Порт  = 1812
18 авг 10:22:09 server1 radiusd[389286]: [0]:connect_to_LDAP_server:Расположение базы данных (где находятся
 данные)=LDAP.
18 авг 10:22:09 server1 radiusd[389286]: [0]:connect_to_LDAP_server:
Имя сервера LDAP = server1.austin.ibm.com.
18 авг 10:22:09 server1 radiusd[389286]: [0]:connect_to_LDAP_server:
```

```
Порт сервера LDAP=389.
18 авг 10:22:09 server1 radiusd[1081472]: [0]:Процесс учета запущен:
Pid= 389286 Порт = 1813
18 авг 10:22:10 server1 radiusd[549080]: [0]:Сокет создан [15]
18 авг 10:22:10 server1 radiusd[549080]: [0]:Ограничение сокета
идентификации [15]
18 авг 10:22:10 server1 radiusd[389286]: [0]:Сокет создан [15]
18 авг 10:22:10 server1 radiusd[389286]: [0]:Ограничение сокета учета [15]
18 авг 10:22:15 server1 radiusd[389286]: [1]:*** Запуск Process_Packet() ***
18 авг 10:22:15 server1 radiusd[389286]: [1]:Входящий пакет:
18 авг 10:22:15 server1 radiusd[389286]: [1]:  Код = 4, ИД = 94, Длина = 80
18 авг 10:22:15 server1 radiusd[389286]: [1]:  Идентификатор = 0xC5DBDDFE6EFFFDBD6AE64CA35947DD0F
18 авг 10:22:15 server1 radiusd[389286]: [1]:    Тип =  40, Длина =   6, Значение = 0x00000001
18 авг 10:22:15 server1 radiusd[389286]: [1]:    Тип =  1, Длина =   8, Значение = 0x67656E747931
18 авг 10:22:15 server1 radiusd[389286]: [1]:    Тип =  4, Длина =   6, Значение = 0x00000000
18 авг 10:22:15 server1 radiusd[389286]: [1]:    Тип =  8, Длина =   6, Значение = 0x0A0A0A01
18 авг 10:22:15 server1 radiusd[389286]: [1]:    Тип =  44, Длина =   8, Значение = 0x303030303062
18 авг 10:22:15 server1 radiusd[389286]: [1]:    Тип =  30, Длина =   10, Значение = 0x3132332D34353638
18 авг 10:22:15 server1 radiusd[389286]: [1]:    Тип =  31, Длина =   10, Значение = 0x3435362D31323335
18 авг 10:22:15 server1 radiusd[389286]: [1]:    Тип =  85, Длина =   6, Значение = 0x00000259
18 авг 10:22:15 server1 radiusd[389286]: [1]:Запуск parse_packet()
18 авг 10:22:15 server1 radiusd[389286]: [1]:Код 4, ИД = 94, Порт = 41639 Хост = 10.10.10.10
18 авг 10:22:15 server1 radiusd[389286]: [1]:Acct-Status-Type = Sta
```

Пакеты идентификации уровня 0

```
18 авг 10:06:11 server1 syslog: [0]:Процесс отслеживания [1081460] запущен
18 авг 10:06:11 server1 radiusd[1081460]: [0]:Локальная база данных (AVL) создана.
18 авг 10:06:11 server1 radiusd[1081460]: [0]:Процесс идентификации запущен : Pid= 549076 Порт = 1812
18 авг 10:06:11 server1 radiusd[1081460]: [0]:Процесс идентификации запущен : Pid= 389282 Порт = 18
```

Пакет идентификации уровня 3

```
18 авг 10:01:32 server2 radiusd[389276]: [3]:*** Запуск Process_Packet() ***
18 авг 10:01:32 server2 radiusd[389276]: [3]:Код 1, ИД = 72, Порт = 41638 Хост = 10.10.10.10
18 авг 10:01:32 server2 radiusd[389276]: [3]:authenticate_password_PAP: Неправильный пароль,
пользователю запрещен доступ
18 авг 10:01:32 server2 radiusd[389276]: [3]:Идентификация для пользователя [user_id1]
с помощью IP-адреса [10.10.10.10] не выполнена
18 авг 10:01:32 server2 radiusd[389276]: [3]:ACCESS-REJECT - отправка отклонения для ИД 72 на 10.10.10.10
  (client1.ibm.com)
18 авг 10:01:32 server2 radiusd[389276]: [3]:send_reject() Исходящий пакет:
18 авг 10:01:32 server2 radiusd[389276]: [3]: Код = 3, ИД = 72, Длина = 30
18 авг 10:01:32 server2 radiusd[389276]: [3]:*** Завершение Process_Packet() ***
18 авг 10:01:53 server2 radiusd[389276]: [4]:*** Запуск Process_Packet() ***
18 авг 10:01:53 server2 radiusd[389276]: [4]:Код 1, ИД = 74, Порт = 41638 Хост = 10.10.10.10
18 авг 10:01:53 server2 radiusd[389276]: [4]:authenticate_password_PAP: Пароль проверен,
идентификация пользователя выполнена
18 авг 10:01:53 server2 radiusd[389276]: [4]:Идентификация для пользователя [user_id1] с
помощью IP-адреса [10.10.10.10] выполнена
18 авг 10:01:53 server2 radiusd[389276]: [4]:Права доступа пользователю [user_id1] успешно
предоставлены с помощью IP-адреса [10.10.10.10]
18 авг 10:01:53 server2 radiusd[389276]: [4]:ACCESS-ACCEPT - отправка подтверждения для ИД 74 на адрес 10.10.10.10
  (client1.ibm.com)
18 авг 10:01:53 server2 radiusd[389276]: [4]:send_accept() Исходящий пакет:
18 авг 10:01:53 server2 radiusd[389276]: [4]: Код = 2, ИД = 74, Длина = 31
18 авг 10:01:53 server2 radiusd[389276]: [4]:*** Завершение Process_Packet() **
```

Пакеты идентификации уровня 9

```
18 авг 10:03:56 server1 radiusd[389278]: [1]:*** Запуск Process_Packet() ***
18 авг 10:03:56 server1 radiusd[389278]: [1]:Входящий пакет:
18 авг 10:03:56 server1 radiusd[389278]: [1]:  Код = 1, ИД = 77, Длина = 58
18 авг 10:03:56 server1 radiusd[389278]: [1]:  Идентификатор = 0xE6CB0F9C22BB4E799854E734104FB2D5
18 авг 10:03:56 server1 radiusd[389278]: [1]:    Тип =  1, Длина =   8, Значение = 0x67656E747931
18 авг 10:03:56 server1 radiusd[389278]: [1]:    Тип =  4, Длина =   6, Значение = 0x00000000
18 авг 10:03:56 server1 radiusd[389278]: [1]:    Тип =  2, Длина =   18, Значение = 0x**********
**********************
18 авг 10:03:56 server1 radiusd[389278]: [1]:    Тип =   7, Длина =   6, Значение = 0x00000001
18 авг 10:03:56 server1 radiusd[389278]: [1]:Запуск parse_packet()
18 авг 10:03:56 server1 radiusd[389278]: [1]:Код 1, ИД = 77, Порт = 41638 Хост = 10.10.10.10
18 авг 10:03:56 server1 radiusd[389278]: [1]:User-Name = "user_id1"
```

```
18 авг 10:03:56 server1 radiusd[389278]: [1]:NAS-IP-Address = 10.10.10.10
18 авг 10:03:56 server1 radiusd[389278]: [1]:Framed-Protocol = PPP
18 авг 10:03:56 server1 radiusd[389278]: [1]:Завершение parse_packet()
18 авг 10:03:56 server1 radiusd[389278]: [1]:Проверка Message-Authenticator
18 авг 10:03:56 server1 radiusd[389278]: [1]:Атрибут Message-Authenticator успешно проверен
18 авг 10:03:56 server1 radiusd[389278]: [1]:Выполнение функции proxy_request_needed()
18 авг 10:03:56 server1 radiusd[389278]: [1]:Proxy не включен
18 авг 10:03:56 server1 radiusd[389278]: [1]:Имя пользователя = [user_id1]
18 авг 10:03:56 server1 radiusd[389278]: [1]:IP-адрес клиента = [10.10.10.10]
18 авг 10:03:56 server1 radiusd[389278]: [1]:Выполнение parse_for_login( user_id1 )
18 авг 10:03:56 server1 radiusd[389278]: [1]:User_id после удаления префикса  = [user_id1]
18 авг 10:03:56 server1 radiusd[389278]: [1]:User_id после удаления суффикса = [user_id1]
18 авг 10:03:56 server1 radiusd[389278]: [1]:Выполнение функции rad_authenticate()
18 авг 10:03:56 server1 radiusd[389278]: [1]:Получен запрос на идентификацию [client1.austin.ibm.com]
18 авг 10:03:56 server1 radiusd[389278]: [1]:Вызов get_ldap_user() для загрузки сведений о пользователе LDAP
18 авг 10:03:56 server1 radiusd[389278]: [1]:get_ldap_user:ИД пользователя LDAP: user_id1.
18 авг 10:03:56 server1 radiusd[389278]: [1]:get_ldap_user:LDAP max_login_cnt:2.
18 авг 10:03:56 server1 radiusd[389278]: [1]:get_ldap_user:LDAP EAP_type: 4.
18 авг 10:03:56 server1 radiusd[389278]: [1]:get_ldap_user:LDAP passwordexpiredweeks: 9.
18 авг 10:03:56 server1 radiusd[389278]: [1]:get_ldap_active_sessions: число свободных записей= 2.
18 авг 10:03:56 server1 radiusd[389278]: [1]:get_ldap_active_session:dn retrieved=
radiusuniqueidentifier=user_id11,ou=radiusActiveUsers,cn=aixradius.
18 авг 10:03:56 server1 radiusd[389278]: [1]:Выполнение процедуры get_client_secret для ip-адреса:10.10.10.10
18 авг 10:03:56 server1 radiusd[389278]: [1]:Найден NAS-IP = [10.10.10.10]
18 авг 10:03:56 server1 radiusd[389278]: [1]:Найден общий шифр.
18 авг 10:03:56 server1 radiusd[389278]: [1]:authenticate_password_PAP: Правильный пароль, идентификация
пользователя выполнена успешно
18 авг 10:03:56 server1 radiusd[389278]: [1]:is_ldap_pw:срок действия пароля пользователя не истек
18 авг 10:03:56 server1 radiusd[389278]: [1]:Идентификация для пользователя [user_id1] с помощью
IP-адреса [10.10.10.10] выполнена успешно
18 авг 10:03:56 server1 radiusd[389278]: [1]:Выполнение процедуры rad_authorize().
18 авг 10:03:56 server1 radiusd[389278]: [1]:Внутри процедура read_authorize_policy для файла:
  /etc/radius/authorization/user_id1.policy.
18 авг 10:03:56 server1 radiusd[389278]: [1]:Внутри процедура read_authorize_file для файла:
  /etc/radius/authorization/user_id1.policy.
18 авг 10:03:56 server1 radiusd[389278]: [1]:Не удалось открыть файл /etc/radius/authorization/user_id1.policy.
  Файл не найден.
18 авг 10:03:56 server1 radiusd[389278]: [1]:Ошибка при чтении файла стратегии:
/etc/radius/authorization/user_id1.policy.
18 авг 10:03:56 server1 radiusd[389278]: [1]:rad_authorize:список стратегий по умолчанию и список
стратегий пользователей не содержат записей.
18 авг 10:03:56 server1 radiusd[389278]: [1]:Выполнение процедуры create_def_copy().
18 авг 10:03:56 server1 radiusd[389278]: [1]:Копия главного списка предоставления прав доступа создана успешно.
18 авг 10:03:56 server1 radiusd[389278]: [1]:Внутри процедура read_authorize_file для файла:
  /etc/radius/authorization/user_id1.auth.
18 авг 10:03:56 server1 radiusd[389278]: [1]:Не удалось открыть файл /etc/radius/authorization/user_id1.auth.
  Файл не найден.
18 авг 10:03:56 server1 radiusd[389278]: [1]:rad_authorize:список предоставления прав доступа и
список пользователей не содержат записей.
18 авг 10:03:56 server1 radiusd[389278]: [1]:Предоставление прав доступа пользователю [user_id1] с
помощью IP-адреса [10.10.10.10] выполнено успешно
18 авг 10:03:56 server1 radiusd[389278]: [1]:ACCESS-ACCEPT - отправка подтверждения для ИД 77 на 10.10.10.10
  (client1.austin.ibm.com)
18 авг 10:03:56 server1 radiusd[389278]: [1]:Выполнение функции proxy_response_needed()
18 авг 10:03:56 server1 radiusd[389278]: [1]:Proxy не включен
18 авг 10:03:56 server1 radiusd[389278]: [1]:Выполнение процедуры get_client_secret для ip-адреса:10.10.10.10
18 авг 10:03:56 server1 radiusd[389278]: [1]:Найден NAS-IP = [10.10.10.10]
18 авг 10:03:56 server1 radiusd[389278]: [1]:Найден общий шифр.
18 авг 10:03:56 server1 radiusd[389278]: [1]:send_accept() Исходящий пакет:
18 авг 10:03:56 server1 radiusd[389278]: [1]:  Код = 2, ИД = 77, Длина = 31
18 авг 10:03:56 server1 radiusd[389278]: [1]:send_accept() Исходящий пакет:
18 авг 10:03:56 server1 radiusd[389278]: [1]:  Код = 2, ИД = 77, Длина = 31
18 авг 10:03:56 server1 radiusd[389278]: [1]:  Идентификатор = 0xCCB2B645BBEE86F5E4FC5BE24E904B2A
18 авг 10:03:56 server1 radiusd[389278]: [1]:    Тип =  18,  Длина =   11, Значение = 0x476F6F646E65737321
18 авг 10:03:56 server1 radiusd[389278]: [1]:*** Завершение Process_Packet() ***
18 авг 10:04:18 server1 radiusd[389278]: [2]:*** Запуск Process_Packet() ***
18 авг 10:04:18 server1 radiusd[389278]: [2]:Входящий пакет:
18 авг 10:04:18 server1 radiusd[389278]: [2]:  Код = 1, ИД = 79, Длина = 58
18 авг 10:04:18 server1 radiusd[389278]: [2]:  Идентификатор = 0x774298A2B6DD90D7C33B3C10C4787D41
18 авг 10:04:18 server1 radiusd[389278]: [2]:    Тип =  1, Длина =   8, Значение = 0x67656E747931
18 авг 10:04:18 server1 radiusd[389278]: [2]:    Тип =  4, Длина =   6, Значение = 0x00000000
18 авг 10:04:18 server1 radiusd[389278]: [2]:    Тип =  2, Длина =  18, Значение = 0x*******
```

```
************************
18 авг 10:04:18 server1 radiusd[389278]: [2]:    Тип =   7,  Длина =   6, Значение = 0x00000001
18 авг 10:04:18 server1 radiusd[389278]: [2]:Запуск parse_packet()
18 авг 10:04:18 server1 radiusd[389278]: [2]:Код 1, ИД = 79, Порт = 41638 Хост = 10.10.10.10
18 авг 10:04:18 server1 radiusd[389278]: [2]:User-Name = "user_id1"
18 авг 10:04:18 server1 radiusd[389278]: [2]:NAS-IP-Address = 10.10.10.10
18 авг 10:04:18 server1 radiusd[389278]: [2]:Framed-Protocol = PPP
18 авг 10:04:18 server1 radiusd[389278]: [2]:Завершение parse_packet()
18 авг 10:04:18 server1 radiusd[389278]: [2]:Проверка Message-Authenticator
18 авг 10:04:18 server1 radiusd[389278]: [2]:Проверка Message-Authenticator выполнена успешно
18 авг 10:04:18 server1 radiusd[389278]: [2]:Выполнение функции proxy_request_needed()
18 авг 10:04:18 server1 radiusd[389278]: [2]:Proxy не включен
18 авг 10:04:18 server1 radiusd[389278]: [2]:Username = [user_id1]
18 авг 10:04:18 server1 radiusd[389278]: [2]:IP-адрес клиента = [10.10.10.10]
18 авг 10:04:18 server1 radiusd[389278]: [2]:Выполнение parse_for_login( user_id1 )
18 авг 10:04:18 server1 radiusd[389278]: [2]:User_id после удаления префикса = [user_id1]
18 авг 10:04:18 server1 radiusd[389278]: [2]:User_id после удаления суффикса = [user_id1]
18 авг 10:04:18 server1 radiusd[389278]: [2]:Выполнение функции rad_authenticate()
18 авг 10:04:18 server1 radiusd[389278]: [2]:Получен запрос на идентификацию для [client1.austin.ibm.com]
18 авг 10:04:18 server1 radiusd[389278]: [2]:Вызов get_ldap_user() для получения пользовательских данных LDAP
18 авг 10:04:18 server1 radiusd[389278]: [2]:get_ldap_user:ИД пользователя LDAP: user_id1.
18 авг 10:04:18 server1 radiusd[389278]: [2]:get_ldap_user:LDAP max_login_cnt:2.
18 авг 10:04:18 server1 radiusd[389278]: [2]:get_ldap_user:LDAP EAP_type: 4.
18 авг 10:04:18 server1 radiusd[389278]: [2]:get_ldap_user:LDAP passwordexpiredweeks: 9.
18 авг 10:04:18 server1 radiusd[389278]: [2]:get_ldap_active_sessions:число свободных записей= 2.
18 авг 10:04:18 server1 radiusd[389278]: [2]:get_ldap_active_session:dn retrieved=
  radiusuniqueidentifier=user_id11, ou=radiusActiveUsers, cn=aixradius.
18 авг 10:04:18 server1 radiusd[389278]: [2]:Выполнение процедуры
get_client_secret для ip-адреса:10.10.10.10
18 авг 10:04:18 server1 radiusd[389278]: [2]:Найден NAS-IP = [10.10.10.10]
18 авг 10:04:18 server1 radiusd[389278]: [2]:Найден общий шифр.
18 авг 10:04:18 server1 radiusd[389278]: [2]:authenticate_password_PAP:
Пароли не совпадают, пользователь отклонен
18 авг 10:04:18 server1 radiusd[389278]: [2]:Идентификация пользователя
[user_id1] с помощью IP-адреса [10.10.10.10] не выполнена
18 авг 10:04:18 server1 radiusd[389278]: [2]:ACCESS-REJECT - отправка отклонения для ИД 79 на 10.10.10.10
 (client1.austin.ibm.com)
18 авг 10:04:18 server1 radiusd[389278]: [2]:Выполнение функции
proxy_response_needed()
18 авг 10:04:18 server1 radiusd[389278]: [2]:Proxy не включен
18 авг 10:04:18 server1 radiusd[389278]: [2]:Выполнение процедуры
get_client_secret для ip-адреса:10.10.10.10
18 авг 10:04:18 server1 radiusd[389278]: [2]:Найден NAS-IP = [10.10.10.10]
18 авг 10:04:18 server1 radiusd[389278]: [2]:Найден общий шифр.
18 авг 10:04:18 server1 radiusd[389278]: [2]:send_reject() Исходящий пакет:
18 авг 10:04:18 server1 radiusd[389278]: [2]:  Код = 3, ИД = 79, Длина = 30
18 авг 10:04:18 server1 radiusd[389278]: [2]:send_reject() Исходящий пакет:
18 авг 10:04:18 server1 radiusd[389278]: [2]:  Код = 3, ИД = 79, Длина = 30
18 авг 10:04:18 server1 radiusd[389278]: [2]:  Идентификатор = 0x05D4865C6EBEFC1A9300D2DC66F3DBE9
18 авг 10:04:18 server1 radiusd[389278]: [2]:    Тип =  18,  Длина =   10, Значение = 0x4261646E65737321
18 авг 10:04:18 server1 radiusd[389278]: [2]:Завершение Leave_Packet()
```

Срок действия пароля

Функция ограничения срока действия паролей позволяет уведомлять клиента RADIUS об устаревании пароля и обновлять пароль по протоколу RADIUS.

Ограничение срока действия паролей предусматривает поддержку четырех дополнительных типов пакетов, а также нового атрибута. Описание пакетов новых типов приведено в словаре AIX. Функция ограничения срока действия паролей должна быть включена.

В некоторых случаях обновление просроченных паролей с помощью протокола RADIUS нежелательно. В файле `radiusd.conf` предусмотрена опция, позволяющая разрешать и запрещать изменение просроченных паролей с помощью RADIUS. По умолчанию изменение паролей запрещено. При необходимости вы можете добавить ответное сообщение `Password_Expired_Reply_Message`, которое будет возвращаться в пакете password-expired. Новые и старые атрибуты паролей должны быть защищены с помощью алгоритма шифрования PAP.

Атрибуты вендоров

Атрибуты вендоров (VSA) определяются вендорами серверов удаленного доступа, как правило, это вендоры аппаратного обеспечения, для дополнительной настройки RADIUS.

Атрибуты вендоров необходимы для предоставления пользователям нескольких типов прав доступа. Их можно применять совместно с атрибутами RADIUS.

Атрибуты вендоров являются необязательными. Однако если для правильной работы аппаратного обеспечения NAS требуются дополнительные атрибуты, вам потребуется добавить их в файл словаря в обязательном порядке.

VSA можно использовать и для дальнейших идентификаций. Кроме **User-Name** и **Password**, для идентификации можно использовать VSA. Файл стратегии прав доступа, расположенный на сервере, содержит список атрибутов из пакета Access-Request, которые должны проверяться для конкретного пользователя. Если пакет не содержит атрибуты, перечисленные в файле пользователя, то серверу сетевого доступа NAS возвращается пакет запрета доступа. Атрибуты вендоров можно указывать в качестве пар атрибут=значение в файле ***ИД-пользователя***`.policy`.

Ниже приведен пример фрагмента раздела VSA из словаря:

```
########################################################################
#                                                                      #
#   В разделе приведены примеры переводов словарей для обработки       #
#   атрибутов, определяемых поставщиком (vsa). Ниже приведен пример    #
#   для "Cisco." Перед тем, как задать пару значение-атрибут для       #
#   поставщика, необходимо определение "VENDOR".                           #
#                                                                      #
#   Пример:                                                            #
#                                                                      #
#   VENDOR          Cisco           9                                  #
#                                                                      #
#   VENDOR:  Указывает, что следующие атрибуты в этой записи относятся #
#            к Cisco.                                  #
#   Cisco :  Задает имя вендора                                        #
#   9     :  ИД вендора, указанный в RFC "Assigned Numbers"            #
#                                                                      #
########################################################################

#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
```

Поддержка ответных сообщений RADIUS

Ответное сообщение - это текст, указанный в файле `radiusd.conf`.

NAS или AP возвращают этот текст конечным пользователям. Это могут быть сообщения о вызове, разрешении доступа или запрете доступа. Это текстовые поля, доступные для чтения, содержимое которых

зависит от конкретной реализации и указывается во время настройки сервера. По умолчанию значения этих атрибутов не заданы. Пи необходимости вы можете настроить произвольное число атрибутов.

RADIUS поддерживает следующие атрибуты ответных сообщений:

- Ответное сообщение разрешения
- Ответное сообщение запрета
- Ответное сообщение CHAP
- Ответное сообщение об истечении срока действия пароля

Указанные атрибуты добавляются в файл конфигурации `radiusd.conf` и загружаются в глобальную структуру конфигурации во время запуска демона. Данные значения можно указать с помощью панелей **Настроить сервер** программы SMIT. Длина каждой строки не может превышать 256 символов.

Эта функция реализована следующим образом:

1. При запуске демон **radiusd** считывает файл `radiusd.conf` и устанавливает атрибуты ответных сообщений.
2. При получении пакета запроса доступа выполняется идентификация пользователя.
3. Если в результате идентификации возвращается разрешение доступа, то выбирается ответное сообщение разрешения. Данное сообщение, если оно указано, возвращается в пакете разрешения доступа.
4. Если идентификация не выполнена, выбирается ответное сообщение запрета и возвращается в пакете запрета доступа.
5. Если необходима дополнительная идентификация, то выбирается сообщение CHAP и возвращается в пакете вызова.

Конфигурация пула IP сервера RADIUS

Сервер RADIUS поддерживает динамическое присвоение IP-адресов из пула IP.

Выделение IP-адресов является частью процесса управления доступом и выполняется после идентификации. Системный администратор должен присвоить каждому пользователю уникальный IP-адрес. Для динамического присвоения IP-адресов на сервере RADIUS предусмотрены три функции:

- Атрибут Framed-Pool
- Использование атрибута вендора
- Создание пула IP на сервере RADIUS

Атрибут Framed-Pool

На сетевом сервере доступа (NAS) определяется пул IP с именем *имя-пула*. Для того чтобы сервер RADIUS мог отправить атрибут **Framed-Pool** в пакет разрешения доступа (атрибут 88 типа), сервер NAS должен быть совместим со стандартом RFC2869. Системный администратор должен настроить NAS и обновить атрибуты управления доступом для пользователей, добавив атрибут **Framed-Pool** в глобальный файл `default.auth` или в файл `user.auth` на сервере RADIUS. Файл словаря на сервере RADIUS включает такой атрибут:

```
ATTRIBUTE    Framed-Pool    88       строка
```

Если NAS не поддерживает несколько пулов адресов, то этот атрибут будет проигнорирован. Пул адресов на сервере NAS содержит список IP-адресов. NAS динамически выбирает один из IP-адресов, определенных в указанном пуле, и присваивает его пользователю.

Атрибуты вендоров

Некоторые независимые вендоры ПО (ISV) не работают с атрибутом **Framed-Pool**, но также имеют возможность определения пулов IP-адресов. Сервер RADIUS может использовать эти адресные пулы

посредством модели атрибутов вендоров (VSA). Например, Cisco NAS предусматривает атрибут Cisco-AVPair. Файл словаря на сервере RADIUS включает такой атрибут:

```
VENDOR     Cisco           9
TTRIBUTE  Cisco-AVPair    1       строка
```

При отправке пакета запроса доступа сервер NAS включает этот атрибут с параметром `Cisco-AVPair="ip:addr-pool=`*`имя-пула`*`"`, где *имя-пула* - это имя пула адресов, определенного на сервере NAS. После идентификации запроса и предоставления доступа сервер RADIUS возвращает атрибут в пакете разрешения доступа. После этого NAS может присваивать пользователям IP-адреса из этого пула. Системный администратор должен настроить NAS и обновить атрибуты управления доступом для пользователей, добавив атрибут VSA в глобальный файл `default.auth` или в файл `user.auth` на сервере RADIUS.

Создание пула IP на сервере RADIUS

Сервер RADIUS можно настроить так, чтобы IP-адреса генерировались из пула IP-адресов. IP-адрес возвращается в атрибуте Framed-IP-Address пакета разрешения доступа.

Пул IP-адресов можно задать с помощью интерфейса SMIT. Адреса хранятся в файле `/etc/radius/ippool_def`. *Имена пулов* указываются в файле `etc/radius/clients`. Также системному администратору следует настроить номер порта NAS. Для создания файлов данных демон сервера RADIUS берет информацию из файлов `etc/radius/clients` и `/etc/radius/ippool_def`. Во время работы демона администратор не может изменять или добавлять *имена пулов* или диапазоны IP-адресов до тех пор, пока серверы RADIUS не будут остановлены. При запуске демон RADIUS считывает файл конфигурации (`/etc/radius/radius.conf`) и, если выделение IP-адресов разрешено (`Enable_IP_Pooling=YES`), то устанавливает значение флага глобального выделения IP-адресов (`IP_pool_flag`) равным On. После этого демон проверяет наличие файла `poolname.data`. Если он есть, то демон читает этот файл и сохраняет информацию в общей памяти. Затем он обновляет файл и общую память на основе запросов от клиентов. Если файла нет, то демон создает новый файл на основе информации из файлов `etc/radius/clients` и `/etc/radius/ippool_def`. Максимально возможный размер файла `poolname.data` равен 256 Мб (максимальный размер сегмента в AIX). Если размер файла `poolname.data` превышает 256 Мб, то сервер RADIUS записывает в протокол сообщение об ошибке и завершает работу.

Демон получает сведения о пуле IP из файла `/etc/radius/ippool_def` и сохраняет таблицу IP-адресов для каждого пула в общей памяти. В этой таблице содержатся записи для IP-адреса NAS, порта NAS и флаг IN USE. Демон ведет хэш-таблицу с ключом NAS-IP NAS-port. Когда приходит запрос от нескольких пользователей, UDP ставит этот запрос в очередь, а демон извлекает из него данные об IP-адресе и порте NAS. По этой информации он проверяет, указано ли для этого NAS *имя-пула*, используя сведения из файла `etc/radius/clients`.

Демон пытается получить из пула свободный адрес. Если такой адрес есть, то он с помощью флагов NAS-IP и NAS-port помечается как “in use” (используемый) и возвращается на сервер RADIUS. IP-адрес помещается демоном в атрибут **Framed-IP-Address** и возвращается серверу NAS в пакете разрешения доступа. Файл `poolname.data` также обновляется, синхронизируясь с информацией, содержащейся в общей памяти.

Если пул не существует или в нем закончились свободные адреса, то серверу RADIUS возвращается ошибка. Ошибка `Невозможно выделить IP-адрес` заносится в файл протокола, и сервер NAS отправляет серверу RADIUS пакет запрета доступа.

Коды ошибок следующие:

- NOT_POOLED – Не определен пул для **nas_ip**.
- POOL_EXHAUSTED – Пул для **nas_ip** определен, но в нем нет свободных адресов.

Если запрос на идентификацию приходит от пары NAS и NAS-порт, которой уже присвоен IP-адрес, то демон возвращает предыдущий выделенный адрес из пула, переводя флаг IN USE в состояние Off и очищая в таблице записи NAS-IP-address и NAS-port. После этого из пула выбирается новый IP-адрес.

IP-адрес также возвращается в пул, если сервер RADIUS получает от NAS пакет останова учета. В этом пакете должны содержаться записи NAS-IP-address и NAS-port. Демон обращается к файлу `ippool_mem` в следующих случаях:

- Пришел запрос на получение нового IP-адреса. Значение флага IN USE устанавливается равным true.
- Получен пакет останова учета. IP-адрес освобождается посредством установки значения флага “in use” равным false.

В каждом случае системные вызовы общей памяти обеспечивают синхронность информации в файлах `poolname.data` и общей памяти. Системный администратор может переключать выделение IP-адресов между `ON` и `OFF` с помощью параметра*Enable_IP_Pooling* в файле конфигурации сервера RADIUS (`radiusd.conf`). Это может пригодиться в ситуации, если IP-адрес присвоен из глобального файла `default.auth` или из локального файла `user.auth`. Для того, чтобы использовать присвоенный IP-адрес, системный администратор должен установить значение `Enable_IP_Pool = NO`.

Пример файла `/etc/radius/ippool_def`, созданного программой SMIT:

Имя пула	**Начало диапазона**	**Конец диапазона**
Floor5	192.165.1.1	192.165.1.125
Floor6	192.165.1.200	192.165.1.253

Ниже приведен пример файла `/etc/radiusclients`, созданного программой SMIT:

NAS-IP	**Общий ключ**	**Имя пула**
1.2.3.4	Secret1	Floor5
1.2.3.5	Secret2	Floor6
1.2.3.6	Secret3	Floor5
1.2.3.7	Secret4	

Обратите внимание, что в примере имя пула для NAS-IP-Address 1.2.3.7 пусто. Для этого NAS пул IP-адресов не создается (даже если значение глобального флага `IP_pool_flag = True`). Когда приходит пакет запроса доступа, сервер RADIUS осуществляет идентификацию и проверку прав доступа. В случае успеха он отправляет пакет разрешения доступа, содержащий статический IP-адрес, определенный в запросе либо в файлах `default.auth` или `user.auth`. Атрибут NAS-Port в данном случае не требуется.

Если организация пула IP `включена`, то системный администратор также указывает статический IP-адрес в глобальном файле `default.auth`, в локальном файле `user.auth`, или в пакете запроса доступа. Сервер RADIUS заменяет этот IP-адрес адресом, выделенным из пула, указанного для этого NAS. Если все IP-адреса пула заняты, то сервер заносит в протокол ошибку (пул исчерпан) и отправляет пакет запрета доступа. Все статические IP-адреса, определенные в файлах `auth`, сервер игнорирует.

Если организация пула IP `включена`, и для NAS указано допустимое имя пула, то когда от этого NAS-IP приходит пакет запроса доступа, в котором не определен порт NAS, то сервер отправляет пакет запрета доступа.

Ниже приведен пример файла `Floor5.data`, созданного демоном:

IP-адрес	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

Ниже приведен пример файла `Floor6.data`, созданного демоном:

IP-адрес	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

Если для указанного NAS необходимо освободить все выделенные IP-адреса (например, при останове NAS), то для инициализации файла *`poolname`*`.data` может потребоваться освободить все IP-адреса из всех пулов. Это можно сделать с помощью следующих пунктов меню программы SMIT:

- Очистить пул IP для клиента
- Полностью очистить пул IP

Панели SMIT для пула IP

В меню Конфигурация клиента выберите **Добавить клиента** и введите **Имя пула** (необязательный параметр). Длина имени не должна превышать 64 символа. Если поле **Имя пула** пусто, то пул IP-адресов создан не будет, и сервер RADIUS будет присваивать IP-адреса, определенные системным администратором в атрибуте проверки прав доступа **Framed-IP-Address**.

Для выбранного **Пула IP** отображаются следующие параметры:

- Показать все пулы IP
- Создать пул IP
- Изменить/Показать характеристики пула IP
- Удалить пул IP
- Очистить пул IP для клиента
- Полностью очистить пул IP

Показать все пулы IP: Этот параметр показывает **Имя пула**, **Начальный IP-адрес диапазона** и **Конечный IP-адрес диапазона**.

Создать пул IP: Этот параметр добавляет имя пула, начальный и конечный адреса диапазона. Эти данные добавляются в конец файла `ippool_def`. При этом проверяется наличие дублированных имен пулов и пересекающихся диапазонов адресов. Эта операция может быть выполнена только если демоны сервера RADIUS не запущены.

Изменить/Показать характеристики пула IP: Этот параметр показывает во всплывающем окне список имен пулов. В этом окне выбирается имя конкретного пула. После этого отображается панель для выбранного пула. При нажатии Enter обновляются данные об этом пуле в файле `ippool_def`. Эта операция может быть выполнена только если демоны сервера RADIUS не запущены.

Удалить пул IP: Выбор этого параметра отображает список доступных имен пулов. После выбора имени появится окно с запросом **Вы уверены?**, в котором следует подтвердить удаление выбранного пула. Для удаления пула из файла `ippool_def` вызывается сценарий **rmippool**. Эта операция может быть выполнена только если демоны сервера RADIUS не запущены.

Очистить пул IP для клиента: Этот параметр устанавливает значение **IN-USE** равным `0` для IP-адресов, относящихся к NAS. Это означает, что все IP-адреса для этого NAS становятся свободными. Эта операция может быть выполнена только если демоны сервера RADIUS не запущены.

Полностью очистить пул IP: При выборе этого параметра появляется окно с запросом **Вы уверены?**, в котором необходимо подтвердить очистку всего файла `ippool_mem`. Эта операция может быть выполнена только если демоны сервера RADIUS не запущены.

Панели SMIT RADIUS

Поля, помеченные звездочкой (*), являются обязательными для заполнения в процессе настройки сервера RADIUS с помощью программы SMIT.

Команда быстрого доступа SMIT:

```
smitty radius
```

Ниже показано Главное меню RADIUS:

```
       Сервер RADIUS

Настроить сервер
Настроить клиентов
Настроить пользователей
Настроить правила Proxy
Расширенная конфигурация сервера
Запустить демоны сервера RADIUS
Остановить демоны сервера RADIUS
```

Ниже приведен пример панели Настроить сервер RADIUS, предусмотренной в программе SMIT:

```
Настроить сервер
  Каталог RADIUS                                        /etc/radius
* Расположение базы данных                          [локальная]       +
  Имя файла локальной базы данных AVL                   [dbdata.bin]
  Уровень отладки                                  [9]                      +#
  Локальный учет                                   [ON]                     +
  Каталог локального учета                 [/var/radius/data/accou>
  Ответные сообщения о разрешении доступа               []
  Ответные сообщения о запрете доступа                  []
  Ответные сообщения о вызове                 []
  Ответные сообщения об устаревании пароля    []
  Поддержка продления просроченных паролей    [NO]                     +
  Требовать идентификатор сообщения           [NO]                     +
*Номер порта идентификации                              [1812]
*Номер порта учета                                      [1813]
  Имя сервера LDAP                                      []
  Номер порта сервера LDAP                    [389]       #
  Отличительное имя администратора сервера LDAP [cn=root]
  Пароль администратора сервера LDAP                    []
  Базовое отличительное имя LDAP                        [cn=aixradius]
  Максимальный размер LDAP                    [0]         #
  Ограничение на пересылку LDAP               [0]         #
  Ограничение по времени LDAP                 [10]        #
  Уровень отладки LDAP                        [0]         +#
  Proxy разрешен                              [OFF]                    +
  Таблица использования Proxy                 [OFF]       +
  Имя области Proxy                                     []
  Разделители префикса Proxy                            [$/]
  Разделители суффикса Proxy                            [@.]
  Удалять транзитные области Proxy            [NO]                     +
  Число попыток Proxy                         [2]                       #
  Тайм-аут Proxy                              [30]                      #
  Проверять ограничения на вход в систему UNIX [OFF]                     +
  Поддержка пула IP                           [OFF]
          +
  Посылать идентификатор сообщения для ACCEPT [ON]                      +
  Максимальное число нитей сервера RADIUS     [15]        #
  Тайм-аут преобразования EAP (в секундах)    [30]        #
  Включить EAP-TLS                            [ON]                     +
  Требуемые параметры для EAP-TLS
       Путь к библиотеке OpenSSL               [/opt/freeware/lib/libs>
       Список алгоритмов шифрования OpenSSL                  [ALL:!ADH:RC4+RSA:+SSLv>
       Корневой каталог CA (полное имя)         [/etc/radius/tls]
       Корневой сертификат CA (полное имя)       [/etc/radius/tls/radius>
       Сертификат сервера RADIUS (полное имя) [/etc/radius/tls/cert-s>
       Секретный ключ сервера RADIUS (полное имя) [/etc/radius/tls/cert-s>
       CRL сервера RADIUS (полное имя)          []
```

Для просмотра подробной справки SMIT по всем полям и опциям нажмите клавишу **F1**.

Генератор случайных чисел

Случайные числа необходимы для создания поля Идентификатор пакета RADIUS.

Генератор случайных чисел должен быть качественным, поскольку злоумышленник может отправить серверу RADIUS запрос, ответ на который заранее просчитан, и использовать этот ответ для получения несанкционированного доступа. Для создания псевдослучайных чисел сервер AIX RADIUS использует расширение ядра **/dev/urandom**. Данное расширение с помощью псевдодрайвера устройств собирает информацию о состоянии аппаратных источников. Тестирование NIST подтвердило соответствующий случайный характер получаемых чисел.

Поддержка глобализации

Команда RADIUS **raddbm** и панели программы SMIT поддерживают глобализацию. Для обеспечения этой функции применяются стандартные вызовы API глобализации AIX.

Связанная информация

Команды: **installp**,**mkuser** и **raddbm**

Предотвращение вторжений AIX

Функция предотвращения вторжений AIX автоматически обнаруживает неуместные, несанкционированные и прочие данные, потенциально опасные для системы.

В следующем разделе рассматриваются различные способы обнаружения вторжений, предусмотренные в AIX.

Связанная информация

Команды: **chfilt**, **ckfilt**, **expfilt**, **genfilt**, **impfilt**, **lsfilt**, **mkfilt**, **mvfilt**, **rmfilt**.

Обнаружение вторжений

Обнаружение вторжений предусматривает отслеживание и анализ событий, происходящих в системе, для обнаружения и пресечения несанкционированного доступа к системе. В AIX обнаружение несанкционированного доступа и попыток такого доступа выполняется путем отслеживания определенных действий и применения к ним правил фильтрации.

Примечание: Для применения функции обнаружения вторжений на хосте необходимо установить набор файлов `bos.net.ipsec`. Технология обнаружения создана на основе стандартных функций защиты протокола IP (IPsec) AIX.

Правила фильтрации с поиском по шаблону:

Поиск по шаблону предусматривает обработку пакетов, передаваемых по сети, с помощью правила фильтрации IPsec. В качестве шаблона фильтра можно применять текстовую строку, шестнадцатеричную строку или файл, содержащий произвольное число шаблонов. В результате обнаружения заданного шаблона в теле какого-либо пакета, выполняется предопределенное действие правила фильтрации.

Правила фильтрации с поиском по шаблону применимы только к входящим пакетам. Для добавления правила фильтрации в таблицу правил фильтрации применяется команда **genfilt**. Правила фильтрации, созданные с помощью этой команды, называются ручными правилами фильтрации. Для включения и выключения правил фильтрации применяется команда **mkfilt**. Кроме того, команда **mkfilt** позволяет управлять ведением протокола фильтра.

В файле шаблона можно указать построчный список текстовых или шестнадцатеричных шаблонов. Правила фильтрации с поиском по шаблону рекомендуется применять для защиты от вирусов, переполнений буферов и прочих вторжений в защиту сети.

Слишком широкое применение правил фильтрации с поиском по шаблону с большим числом шаблонов может привести к заметному снижению производительности системы. Рекомендуется максимально ограничивать область применения этих правил. Например, если известно, что вирус поражает только данные, передаваемые командой **sendmail**, соответствующее правило фильтрации следует указать только для целевого порта SMTP 25 команды **sendmail**. Такой подход позволяет передавать остальные данные без задержки, связанной с поиском по шаблону.

Команда **genfilt** поддерживает формат шаблонов, применяемый в некоторых версиях ClamAV.

Информация, связанная с данной:

Команда genfilt

Команда mkfilt

Веб-сайт ClamAV

Типы шаблонов:

Предусмотрено три основных типа шаблонов: текстовые, шестнадцатеричные и файловые. Правила фильтрации с поиском по шаблону применимы только к входящим пакетам.

Текстовый шаблон

Текстовый шаблон фильтрации - это строка а формате ASCII, которая выглядит примерно следующим образом:

```
GET /../../../../../../../../
```

Шестнадцатеричный шаблон

Шестнадцатеричный шаблон выглядит примерно следующим образом:

```
0x33c0b805e0cd16b807e0cd1650558becc7460200f05d0733ffb8c800b9fffff3abb00150
e670e47132c0e67158fec03c8075f033c033c9b002fa99cd26fb4183f90575f5c3
```

Примечание: Для отличия от текстового шаблона в начале шестнадцатеричного шаблона указывается префикс 0x.

Файлы с текстовыми шаблонами

В файле шаблона можно указать построчный список текстовых или шестнадцатеричных шаблонов. Примеры файлов шаблонов приведены на Web-сайте http://www.clamav.net.

Правила фильтрации с блокировкой порта и блокировкой хоста:

Правило фильтрации с блокировкой позволяет запретить удаленному хосту или паре удаленный хост/порт доступ в локальную систему.

Правило фильтрации с блокировкой динамически создает эффективное правило, запрещающее доступ удаленного хоста или удаленного хоста, подключенного к конкретному порту, к локальной системе в случае удовлетворения заданных критериев правила.

Поскольку вторжение зачастую сопровождается предварительным сканированием портов, правила фильтрации с блокировкой портов, позволяющие обнаруживать действия, свойственные вторжениям, в особенности эффективны для защиты от них.

Например, если локальных хост не применяет порт сервера времени 37, то удаленный хост, обращающийся к этому порту, может выполнять сканирование портов. Укажите для порта 37 правило фильтрации с блокировкой портов, которое в свою очередь создаст эффективное правило, запрещающее доступ этого хоста на время, заданное в поле **срок действия** правила с блокировкой.

Если в поле **срок действия** указано нулевое значение, то блокировка будет применяться постоянно.

Примечание:

1. Срок действия, указанный для правила фильтрации с блокировкой портов, применим только для эффективного правила, созданного в динамическом режиме.
2. Просмотреть список динамически созданных эффективных правил можно с помощью команды **lsfilt -a**.

Правила фильтрации с блокировкой хоста

В результате удовлетворения критериев правила фильтрации с блокировкой хостов динамически создается эффективное правило, блокирующее поток данных от удаленного хоста на заданное время.

Правила фильтрации с блокировкой порта

В результате удовлетворения критериев правила фильтрации с блокировкой портов динамически создается эффективное правило, блокирующее поток данных от конкретного порта удаленного хоста на заданное время.

Правила фильтрации с учетом состояния:

Фильтры с учетом состояния проверяют такую информацию, как исходные и целевые адреса, номера портов и строку состояния. Последующее применение правил фильтрации IF, ELSE и ENDIF к флагам заголовков позволяет принять соответствующее решение в контексте всего сеанса, а не отдельного пакета или информации из его заголовка.

Правила фильтрации с учетом состояния применимы как к входящим, так и исходящим пакетам. После активации этих правил с помощью команды **mkfilt -u** блоки ELSE проверяются до тех пор, пока не будет выполнено условие правила IF. После выполнения условия IF правила блока IF применяются до тех пор, пока правила фильтрации не будут повторно активированы с помощью команды **mkfilt -u**.

Команда **ckfilt** проверяет формат правил фильтрации с учетом состояния и отображает эти правила в наглядной форме, приблизительно так, как это показано ниже:

```
%ckfilt -v4
Начало правил фильтрации IPv4.
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
```

Правила с ограничением по времени:

Правила с ограничением по времени задают интервал времени в секундах, в течение которого правило фильтрации применяется после активации с помощью команды **mkfilt -v [4|6] -u**.

Для указания срока действия применяется команда **genfilt -e**. Дополнительная информация приведена в описании команд **mkfilt** и **genfilt**.

Примечание: Таймеры неприменимы к правилам IF, ELSE и ENDIF. Если срок действия указан для правила с блокировкой хостов или портов, то таймер применяется только к порожденным ими эффективным правилам. Для самих правил с блокировкой сроки действия неприменимы.

Работа с правилами фильтрации посредством SMIT

Правила фильтрации можно настроить с помощь программы SMIT.

Для настройки правил фильтрации с помощью SMIT выполните следующие действия.

1. В командной строке введите следующую команду: `smitty ipsec4`
2. Выберите **Расширенная конфигурация защиты IP**.
3. Выберите **Настроить правила фильтрации защиты IP**.
4. Выберите **Добавить правило фильтрации защиты IP**.

```
                        Добавить правило фильтрации защиты IP

Введите или выберите значения в полях ввода.
После внесения изменений нажмите клавишу Enter.

[Начало]                                                  [Поля ввода]
* Действие правила                                        [permit]          +
* Исходный IP-адрес                                       []
* Маска исходного IP-адреса                               []
  Целевой IP-адрес                                       []
  Маска целевого IP-адреса                               []
* Применить к маршрутизации отправителя (PERMIT/входящие) [да]              +
* Протокол                                                [все]             +
* Исходный порт / Тип операции ICMP                       [любой]           +
* Номер исходного порта / Тип ICMP                        [0]               #
* Целевой порт / Тип операции ICMP                        [любой]           +
* Номер целевого порта / Тип ICMP                         [0]               #
* Маршрутизация                                           [оба]             +
* Направление                                             [оба]             +
* Управление протоколом                                   [нет]             +
* Управление фрагментацией                                [0]               +
* Интерфейс                                               []                +
  Время истечение срока (секунды)                        []                #
  Тип шаблона                                            [нет]             +
  Шаблон  / Файл шаблона                                 []
  Описание                                               []


Где для атрибута "Тип шаблона" допустимы следующие значения:
  x   нет                                                                   x#
  x   шаблон                                                                x
  x   файл                                                                  x
  x   шаблоны защиты от вирусов
```

Для поля action допустимы следующие значения: `permit`, `deny`, `shun_host`, `shun_port`, `if`, `else`, `endif`.

Если указан файл шаблона, он должен быть доступен для чтения при активации правил фильтрации с помощью команды **mkfilt -a**. Правила фильтрации хранятся в базе данных /etc/security/ipsec_filter.

Эксперт безопасности AIX

В Эксперт безопасности AIX предусмотрен центр для управления всеми настройками безопасности (TCP, NET, IPSEC, системы и контроля).

Эксперт безопасности AIX является системной утилитой усиления защиты. Она входит в набор файлов **bos.aixpert**. Эксперт безопасности AIX предоставляет простые настройки меню для Высокого, Среднего и Низкого уровней защиты, а Стандартные настройки защиты AIX, которые объединяют более 300 параметров настройки защиты, при этом контролируя каждый элемент безопасности для расширенных администраторов. Эксперт безопасности AIX можно использовать для поддержания соответствующего уровня безопасности, не изучая документацию по усилению безопасности и не настраивая каждый элемент безопасности в отдельности.

Эксперт безопасности AIX можно использовать для получения моментальной копии настроек безопасности. Эту моментальную копию можно использовать для задания аналогичной конфигурации защиты в других системах. Это позволяет сэкономить время; кроме того, благодаря этому во всех системах можно задать надлежащую конфигурацию защиты для всех приложений.

Эксперт безопасности AIX можно запустить из SMIT или с помощью команды **aixpert**.

Параметры Эксперт безопасности AIX

Доступны следующие параметры защиты на уровне крупных структурных единиц:

Защита высокого уровня
Защита высокого уровня

Защита среднего уровня
Защита среднего уровня

Защита низкого уровня
Защита низкого уровня

Расширенная защита
Нестандартная пользовательская защита

Стандартные параметры AIX
Системная защита по умолчанию

Отменить защиту
Некоторые параметры Эксперт безопасности AIX можно отменить.

Проверка защиты
Подробный отчет о текущих параметрах защиты

Усиление защиты Эксперта безопасности AIX

При применении усиления защиты, увеличивается безопасность элементов системы или применяется более высокий уровень безопасности системы.

С помощью усиления безопасности можно проверить, что настройки безопасности отвечают вашим требованиям. Можно изменить сотни настроек, чтобы усилить безопасность системы AIX.

Эксперт безопасности AIX предоставляет меню для объединения оптимальных общих настроек безопасности. Эти настройки основаны на обширных исследованиях по обеспечению надлежащего уровня безопасности для систем UNIX. Предоставлены настройки безопасности для различных сред с различными потребностями к уровню безопасности (высокий, средний и низкий уровни безопасности), а опытные администраторы могут задать каждую настройку безопасности в отдельности.

Если задан слишком высокий уровень безопасности, то доступ к каким-либо необходимым службам может быть запрещен. Например, для высокого уровня безопасности отключены **telnet** и **rlogin**, так как пароль для входа в систему отправляется по сети в незашифрованном виде. Если установлен слишком низкий уровень безопасности, система может оказаться легко уязвимой. Поскольку каждое предприятие имеет свои требования к системе безопасности, настройки, заданные с помощью, высокого, среднего и низкого уровня безопасности могут не удовлетворять требованиям отдельно взятого предприятия, но их удобно взять за основу для настройки системы безопасности.

Практичным подходом к использованию Эксперт безопасности AIX является установка тестовой системы (в приближенной к реальности тестовой среде), сходной с производственной средой, в которой будет происходить ее развертывание. Установите требуемые бизнес-приложения и выполните Эксперт безопасности AIX из GUI. Эксперт безопасности AIX проанализирует запущенную систему в защищенном состоянии. В зависимости от выбранных опций защиты Эксперт безопасности AIX включит защиту от просмотра портов, проверку, заблокирует сетевые порты, которые не используются бизнес-приложениями или другими службами, а также многие другие настройки защиты. После повторного тестирования с установкой этой конфигурации защиты система готова к развертыванию в производственной среде. Кроме того, Эксперт безопасности AIX XML-файл, определяющий политику защиты или конфигурацию этой системы можно использовать для реализации точно такой же конфигурации на подобных системах предприятия.

Дополнительную информацию по усилению системы безопасности можно найти в Специальном издании NIST 800-70, Программа справочных таблиц настройки системы безопасности NIST для IT продуктов.

Защита по умолчанию

Защита по умолчанию (SbD) - это концепция установки минимального пакета программного обеспечения в защищенной конфигурации.

Опция установки AIX Защита по умолчанию (SbD) устанавливает облегченную версию наборов серверных файлов клиента TCP, что исключает наличие уязвимых команд и файлов. Наборы файлов **bos.net.tcp.client** и **bos.net.tcp.server** являются частью пакета установки SbD и содержат все команды и файлы за исключением приложений, которые позволяют передавать пароли по сети в виде незашифрованного текста, например **telnet** и **ftp**. Кроме того, приложения, которые могут быть использованы, например **rsh**, **rcp** и **sendmail**, исключены из наборов файлов SbD.

Последним автоматическим процессом в установке SbD является применение настроек конфигурации высокого уровня защиты Эксперт безопасности AIX. Оно производится с помощью команды **aixpert** в сценарии `/etc/firstboot`: `/usr/sbin/aixpert -f /etc/security/aixpert/core/SbD.xml -p 2>/etc/security/aixpert/log/firstboot.log`

Вывести компьютер из режима SbD можно путем изменения значения переменной ODM *SbD_STATE* на sbd_disable и повторной установки наборов файлов **bos.net.tcp.client** и **bos.net.tcp.server**, а затем перевода системы на уровень защиты, установленный для нее по умолчанию, с помощью Эксперт безопасности AIX.

Обновление версии и установка с сохранением не приводят к установке системы с опцией SbD. SbD является отдельным пунктом меню установки.

Примечание: После обновления системы в режиме SbD с помощью пакета обновления этот режим будет отключен.

Можно настроить защищенную систему и без использования опции установки SbD. Например, для обычной установки можно настроить опции Эксперт безопасности AIX защиты Высокого, Среднего или Низкого уровня.

Различия между системой, установленной с опцией SbD, и системой, установленной в обычном режиме с использованием конфигурации Эксперт безопасности AIX Защита высокого уровня наилучшим образом видна при проверке команды **telnet**. В обоих случаях команда **telnet** отключена. При установке SbD двоичный файл или приложение **telnet** даже не устанавливается на систему.

При установке SbD не будут установлены либо будут отключены следующие службы. Если некоторые из этих служб не установлены в системе, о к ним невозможно получить доступ или запустить их из системы. Если эти команды и программы необходимы, то не используйте опцию установки SbD. Кроме того, не используйте опцию установки SbD, если какие-либо сценарии, удаленные программы или зависимые наборы файлов требуют наличия этих команд и программ.

Служба	Программа	Аргументы
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

Служба	Программа	Аргументы
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
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

Также имеется несколько функций в консоли IBM Systems Director для AIX, в том числе портлет HealthMetrics, которые недоступны, когда операционная система AIX работает в режиме SbD. Эти функции можно включить путем установки наборов файлов, требуемых для запуска функции.

Распределение политики защиты с помощью LDAP

LDAP можно использовать для распределения файлов конфигурации Эксперт безопасности AIX в формате XML. Для копирования настроек безопасности из одной системы в другую можно использовать Эксперт безопасности AIX. Это позволяет использовать одни и те же настройки защиты в различных системах. Такая согласованность может понизить степень уязвимости системы.

Рекомендуется использовать Эксперт безопасности AIX для настройки отдельной отдельной системы и установки уровня защиты в соответствии с корпоративными политиками защиты и среды, в которой будет работать система. Эти настройки сохраняются в файле `/etc/security/aixpert/core/appliedaixpert.xml`. Впоследствии его можно перенести на настроенный защищенный сервер LDAP. Другие системы, которые могут соединяться с этим сервером LDAP, будут автоматически находить этот XML-файл конфигурации с помощью команды **aixpertldap**.

На любом существующем сервере LDAP можно обновить схему aixpert для распределения XML-файлов конфигурации между всеми подключенными клиентами. Если на сервере LDAP схема aixpert не обновлена, то схему aixpert на сервере LDAP следует обновить с помощью следующей команды: `ldapmodify -c -D <bindDN> -w <bindPwd> -i /etc/security/ldap/sec.ldif`. После обновления схемы aixpert на сервере LDAP клиенты могут располагать свои XML-файлы конфигурации на LDAP с помощью опции -u команды **aixpertldap**. Эти файлы конфигурации следует обновлять вручную.

Примечание: Эта функция базируется на модели защиты, используемой LDAP. Пользователи с правами записи на LDAP могут изменять данные, которые загружены пользователями другого компьютера. Подобным образом, если в системе защиты клиента LDAP существуют уязвимые места, то их можно использовать для того чтобы прочесть и проанализировать состояние защиты других клиентов LDAP путем чтения XML-файлов конфигурации Эксперт безопасности AIX, связанных с клиентом.

Например, файл `appliedaixpert.xml` можно сохранить на сервере LDAP под именем **BranchOfficeSecurityProfile**. Или файл `appliedaixpert.xml` с другой конфигурацией можно сохранить под именем **InternetDirectAttachedSystemsProfile**. Поскольку другие системы, которые могут соединяться с LDAP, настраиваются с помощью Эксперт безопасности AIX, эти профили защиты автоматически

становятся доступными в виде опций меню. Это позволяет системному администратору выбрать профиль защиты, наиболее приемлемый для среды в соответствии с указаниями корпоративной политики защиты.

После этого Эксперт безопасности AIX используется для защиты системы. Полный перечень настроек защиты, установленных в системе, хранится в файле `/etc/security/aixpert/core/appliedaixpert.xml`. Этот файл является политикой защиты для данной системы. Политика защиты проверяется при использовании опции Эксперт безопасности AIX Проверять защиту. Эту политику защиты также можно скопировать и применить в других системах, что обеспечит соответствие систем защиты во всей информационной среде корпорации. Существуют два способа копирования политики защиты в другие системы: вручную или с помощью LDAP.

Копия стратегии защиты Эксперта безопасности AIX

Эксперт безопасности AIX можно использовать для копирования стратегии защиты из одной системы в другую.

Можно запустить Эксперт безопасности AIX в одной системе и применить ту же самую стратегию защиты в другой. Например, Боб хочет применить Эксперт безопасности AIX в шести своих системах AIX. Он устанавливает высокий, средний, низкий уровень безопасности AIX или задает стандартные настройки безопасности для одной системы (системы Альфа). Он проверяет эту систему на совместимость с программным обеспечением. Если эти настройки удовлетворяют его требованиям, он может применить те же настройки в других системах AIX. Он может копировать настройки из системы Альфа в другие системы, в которых необходимо установить те же настройки, скопировав файл `/etc/security/aixpert/core/appliedaixpert.xml` из системы Альфа в другие системы.

Примечание: Не копируйте этот файл в тот же самый каталог и с тем же именем в другую систему, поскольку команда **aixpert** перезапишет `/etc/security/aixpert/core/appliedaixpert.xml` при реализации стратегии защиты.

Вместо этого скопируйте стратегию защиты системы Альфа в каталог `/etc/security/aixpert/custom/`. Это позволит другой системе обнаружить и применить стратегию защиты системы Альфа с помощью GUI управления системой Эксперт безопасности AIX или напрямую с использованием команды **aixpert**.

Например, если стратегия защиты системы Альфа appliedaixpert.xml была помещена в другую систему как `/etc/security/aixpert/custom/`*`AlphaPolicy`*, то команда `aixpert -f /etc/security/aixpert/custom/`*`AlphaPolicy`* немедленно применит эту стратегию защиты, и в этой системе будет установлена та же конфигурация защиты, что и в системе Альфа. Кроме того, когда стратегия защиты системы Альфа находится в этом каталоге, она видна и может быть применена в других системах с помощью консоли управления системой через с путем Эксперт безопасности AIX -> Обзор и задачи -> Опции пользователя -> AlphaPolicy.

Пользовательская стратегия защиты с применением пользовательских правил Эксперта безопасности AIX AIX в формате XML

Для настройки уникальных стратегий защиты можно использовать XML-файлы.

Эксперт безопасности AIX динамически распознает эти XML-файлы. Все создаваемые XML-файлы стратегий защиты должны размещаться в каталоге `/etc/security/aixpert/custom/` с файлом описания. Поэтому при входе в Эксперт безопасности AIX с помощью графического интерфейса консоли в полной мере реализуется обширный набор графических функций XML в aixpert DTD.

Применяется следующая DTD:

```
<?xml version='1.0'?>

<!--START-->
```

```
<!ELEMENT AIXPertSecurityHardening (AIXPertEntry+)>

<!-- AIXPertEntry должна содержать только по одному экземпляру следующих элементов. -->

<!ELEMENT AIXPertEntry (AIXPertRuleType,
  AIXPertDescription, AIXPertPrereqList, AIXPertCommand,
  AIXPertArgs,AIXPertGroup)>

<!-- Имя AIXPertEntry должно быть уникальным. -->

<!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)*>
```

Имя AIXPertEntry является уникальным в XML-файле. Это имя будет именем графической кнопки при просмотре файла с помощью системной консоли с с использованием пути Эксперта безопасности AIX -> Обзор и задачи -> Опции пользователя -> <файл xml=""></xml>.

<!ELEMENT AIXPertRuleType EMPTY>
Этот XML-файл должен быть указан как пользовательский.

<!ATTLIST AIXPertRuleType type (LLS|MLS|HLS|DLS|SCBPS|Prereq|Custom) "DLS"
Этот XML-файл должен быть указан как пользовательский.

<!ELEMENT AIXPertDescription (#PCDATA)>
В время просмотра с помощью указанного выше графического интерфейса текст описания будет показан в виде всплывающего окна при наведении мыши на эту кнопку.

<!ELEMENT AIXPertPrereqList (#PCDATA)>
Для этого правила можно указать предварительное правило. Предварительное правило должно возвратить 0 перед тем, как aixpert применит данное правило. Если XML-файл просматривается с помощью графического интерфейса, то это правило будет отключено при несоблюдении предварительного правила. Для создания предварительного правила элемент AIXPertRuleType должен иметь значение 'Prereq'.

Поле AIXPertDescription предварительного правила должно описывать, что требуется сделать для соответствия предварительному правилу. Если пользовательское правило неактивно по причине несоблюдения одного из его предварительных правил, то пользователю будет показано всплывающее окно предварительного правила с описанием, объясняющим пользователю, что он должен сделать для исправления предварительного условия.

<!ELEMENT AIXPertCommand (#PCDATA)>
Этот элемент должен указывать полный путь и команду, которую выполнит aixpert для данного правила защиты, например `/usr/bin/ls`.

<!ELEMENT AIXPertArgs (#PCDATA)*>
Этот элемент должен содержать аргументы указанной выше команды, например -l

<!ELEMENT AIXPertGroup (#PCDATA)*>
При просмотре правил в графических интерфейсах можно группировать наборы правил aixpert. Например, в наборе правил для каждого можно указать "Network Security" в качестве имени AIXPertGroup.

Строгая проверка сложности паролей

Эта функция AIX проверяет сложность паролей при их смене. При выборе данной опции в Эксперт безопасности AIX эта дополнительная проверка паролей проводится при выборе или смене пароля пользователем. Такая проверка защищает при использовании слов английского языка и 1000 наиболее употребительных в США имен по данным последней переписи в США.

Цели контроля COBIT, поддерживаемые Экспертом безопасностиAIX

Кроме настроек защиты по умолчанию и дополнительных настроек AIX высокого, среднего и низкого уровня Эксперт безопасности AIX поддерживает уровень защиты в соответствии с требованиями SOX-COBIT к практике защиты.

Для защиты инвесторов путем повышения точности и надежности финансовой информации, разглашаемой корпорациями, Конгресс Соединенных Штатов Америки принял 'Закон Сарбейнса-Оксли от 2002 г.' Функция целей контроля COBIT поможет системным администраторам настраивать свои системы ИТ, обслуживать и контролировать их в соответствии с этим законом. Помощник по настройке SOX вызывается с использованием командной строки aixpert. Компонент помогает в работе с разделом SOX 404 закона Сарбейнса-Оксли, а Помощник по настройке SOX Эксперта безопасности AIX автоматически реализует настройки защиты, обычно связанные с практикой COBIT для раздела SOX 404, Внутренний контроль. Кроме того, Эксперт безопасности AIX реализует функцию контроля по SOX, которая сообщает проверяющему, настроена ли система таким образом на данный момент. Функция позволяет автоматизировать настройку системы, способствуя соответствию ИТ закону SOX и автоматизации контроля.

Поскольку SOX не предоставляет руководства по соответствию ИТ разделу 404, отрасль ИТ использует существующее руководство, расположенное по адресу www.isaca.org/. А точнее, руководство по ИТ, включенное в Цели контроля информационных и смежных технологий (COBIT).

Эксперт безопасности AIX поддерживает следующие цели контроля:

- Применение стратегии управления паролями
- Отчеты о нарушениях и мерах защиты
- Предупреждение появления вредоносных программ, их распознавание и исправление, а также несанкционированные программы
- Архитектура брандмауэров и соединение с сетями общего пользования

Эксперт безопасности AIX не поддерживает все атрибуты, указанные для каждой цели контроля. Поддерживаемые атрибуты и соответствующие цели контроля описаны в следующих таблицах:

Применение стратегии управления паролями

Описание	**Настройка защиты**
Максимальный срок действия пароля	maxage=13
Включить историю паролей	histsize=20
Минимальный срок действия пароля	minage=1
Минимальная длина пароля	minlen=8
Содержит как минимум 6 знаков	Minalpha=6
Сходство с прежним паролем	mindiff=4
Предупреждение об истечении срока действия пароля	pwdwarntime=14

Отчет о нарушениях защиты и принятых мерах

Описание	Настройка защиты	Примечания
Контроль включен	да	
Отсутствие входов в систему непосредственно через root	да	
Включить контроль увеличения прав доступа	да	AIXpert вызывает событие контроля USER_SU. Убедитесь, что это событие включено.

Выявление и исправление вредоносных программ

Эксперт безопасности AIX усовершенствует функцию выполнения защищенных программ AIX, чтобы гарантировать, что программы никем не подделаны. Команда **trustchk** проверяет структуру объектов, зарегистрированных в базе данных защищенных программ.

Настройка брандмауэра

Эксперт безопасности AIX включает IPSec и активирует правила фильтра во избежание просмотра портов. Защищенные порты перечислены в следующей таблице:

Служба	Описание
Tcp/11, udp/11	Systat
Tcp/13, udp/13	Дневное время
(RFC 867) Tcp/19, udp/19	Генератор символов
Tcp/25	Простой протокол передачи почты (SMTP)
Tcp/43, udp/43	Who Is (псевдоним)
Tcp/63, udp/63	Whois++
Tcp/67, udp/67	Сервер протокола начальной загрузки (bootps)
Tcp/68, udp/68	Клиент протокола начальной загрузки (bootpc)
Tcp/69, udp/69	Упрощенный протокол передачи файлов.
(tftp) Tcp/79, udp/79	Finger
Tcp/87	Частная абонентская линия
Tcp/110	Почтовый протокол версии 3 (POP3)
Udp/111	SUN Remote Procedure Call
Tcp/113	Служба идентификации (auth)
Udp/123	Протокол сетевого времени
Udp/161	SNMP
Udp/162	SNMPTRAP
Tcp/194	Протокол IRC
Tcp/443	протокол http по TLS/SSL
Tcp/511	PassGo
Tcp/514	Cmd (shell)
Tcp/520	Сервер расширенных имен файлов (efs)
Tcp/540	Uucpd (uucp)
Tcp/546	Клиент DHCPv6

Служба	Описание
Tcp/547	Сервер DHCPv6
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	Зарезервировано

Применение целей контроля по стандарту COBIT с использованием Эксперта безопасности AIX

Для применения в системе уровня SCBPS можно использовать команду **aixpert –l s**. Для этого с помощью включения события **AIXpert_apply** создается протокол контроля. После его включения сообщения о любых сбоях (как предварительных, так и при применении) поступают в **stderr** и подсистему контроля.

Функция проверки соответствия SOX-COBIT, контроля и предварительной проверки

Для проверки соответствия системы SOX-COBIT можно использовать команду **aixpert –c –l s**. Эксперт безопасности AIX проверяет только соответствие поддерживаемым целям контроля. Обо всех нарушениях, выявленных в ходе проверки, сообщается в отчете. По умолчанию все нарушения отправляются в **stderr**.

Ту же команду (**aixpert –c –l s**) можно использовать для создания отчета о проверке соответствия SOX-COBIT. Для создания отчета о проверке установите и включите подсистему проверки. Убедитесь, что событие проверки **AIXpert_check** включено. После установки подсистемы проверки снова выполните команду **aixpert –c –l s**. Команда создаст протокол контроля для каждой цели контроля, которая не достигнута. В поле **Состояние** протокола контроля появится отметка **сбой**. Также в протоколе будет указана причина сбоя, которую можно узнать с помощью опции **-v** команды **auditpr**.

Добавление опции **–p** к команде **aixpert –c –l s** включает в отчет о проверке также успешные цели контроля. В поле состояния этих записей указано **Ok**.

Команда **aixpert -c -l s -p** может использоваться для создания подробного отчета о проверке соответствия SOX-COBIT.

Независимо от того, указана ли опция **–p**, в отчет будет включена итоговая запись. Итоговая запись содержит информацию о количестве обработанных правил, количестве сбоев, (случаи выявленного несоответствия) и уровне защиты, в соответствии с которым проверяется система(в данном случае, им является SCBPS).

Группа правил стратегии паролей Эксперта безопасности AIX

Эксперт безопасности AIX предоставляет особые правила для стратегии паролей.

Надежные стратегии паролей являются неотъемлемыми элементами безопасности системы. Стратегии паролей гарантируют, что пароли сложно угадать (пароли содержат подходящие сочетания букв, цифр и специальных символов), регулярно истекает срок действия паролей, и что пароли не могут быть

использованы повторно после истечения срока действия. В следующей таблице приведены списки правил стратегии паролей для различных настроек безопасности.

Таблица 20. Эксперт безопасности AIX - Правила стратегии паролей

Имя кнопки	Определение	Значение, заданное Эксперт безопасности AIX	Отме- на
Минимальное число символов	Задает соответствующее значение для атрибута **mindiff** в /etc/security/user, который указывает наименьшее число символов, которые должен содержать новый пароль, и которые не содержал прежний.	**Защита высокого уровня** 4 **Защита среднего уровня** 3 **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** Без ограничений	Да
Минимальный срок хранения пароля	Задает соответствующее значение для атрибута **minage** в /etc/security/user, который указывает наименьшее число недель, по прошествии которых можно изменить пароль.	**Защита высокого уровня** 1 **Защита среднего уровня** 4 **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** Без ограничений	Да
Максимальный срок хранения пароля	Задает соответствующее значение для атрибута **maxage** в /etc/security/user, который указывает наибольшее число недель, по прошествии которых можно изменить пароль.	**Защита высокого уровня** 13 **Защита среднего уровня** 13 **Защита низкого уровня** 52 **Стандартные настройки AIX** Без ограничений	Да
Минимальная длина пароля	Задает соответствующее значение для атрибута **minlen** в /etc/security/user, который указывает наименьшую длину пароля.	**Защита высокого уровня** 8 **Защита среднего уровня** 8 **Защита низкого уровня** 8 **Стандартные настройки AIX** Без ограничений	Да
Минимальное число букв в пароле	Присваивает соответствующее значение атрибуту **minalpha** для /etc/security/user, которое устанавливает минимальное количество букв в пароле.	**Защита высокого уровня** 2 **Защита среднего уровня** 2 **Защита низкого уровня** 2 **Стандартные настройки AIX** Без ограничений	Да

Таблица 20. Эксперт безопасности AIX - Правила стратегии паролей (продолжение)

Имя кнопки	Определение	Значение, заданное Экспертом безопасности AIX	Отме- на
Срок сброса пароля	Задает соответствующее значение для атрибута **histexpire** в /etc/security/user, который указывает число недель, по прошествии которых пароль можно сбросить.	**Защита высокого уровня** 13 **Защита среднего уровня** 13 **Защита низкого уровня** 26 **Стандартные настройки AIX** Без ограничений	Да
Наибольшее число одинаковых символов в пароле	Задает соответствующее значение для атрибута **maxrepeats** в /etc/security/user, который указывает наибольшее число одинаковых символов в пароле.	**Защита высокого уровня** 2 **Защита среднего уровня** Никакого действия **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** 8	Да
Повторное использование пароля	Задает соответствующее значение для атрибута **histsize** в /etc/security/user, который указывает число предыдущих паролей, которые нельзя использовать повторно.	**Защита высокого уровня** 20 **Защита среднего уровня** 4 **Защита низкого уровня** 4 **Стандартные настройки AIX** Без ограничений	Да
Срок изменения пароля после истечения срока действия	Задает соответствующее значение для атрибута **maxexpired** в /etc/security/user, который задает наибольшее число недель, после **maxage**, в течение которых пользователь может изменить пароль.	**Защита высокого уровня** 2 **Защита среднего уровня** 4 **Защита низкого уровня** 8 **Стандартные настройки AIX** -1	Да
Минимальное число символов, отличных от букв, в пароле	Задает соответствующее значение для атрибута **minother** в /etc/security/user, который указывает наименьшее число символов, отличных от букв, в пароле.	**Защита высокого уровня** 2 **Защита среднего уровня** 2 **Защита низкого уровня** 2 **Стандартные настройки AIX** Без ограничений	Да

Таблица 20. Эксперт безопасности AIX - Правила стратегии паролей (продолжение)

Имя кнопки	Определение	Значение, заданное Эксперт безопасности AIX	Отме- на
Предупреждение об истечении срока действия пароля	Задает соответствующее значение для атрибута **pwdwarntime** в `/etc/security/user`, который указывает количество дней, по прошествии которых система посылает предупреждение о необходимости смены пароля.	**Защита высокого уровня** 5 **Защита среднего уровня** 14 **Защита низкого уровня** 5 **Стандартные настройки AIX** Без ограничений	Да

Группа определений паролей и система пользовательских групп Эксперта безопасности AIX

Эксперт безопасности AIX предоставляет особые задачи для пользователей, групп и определений паролей.

Таблица 21. Эксперт безопасности AIX - Определения паролей и система пользовательских групп

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отменить
Проверить определения групп	Проверяет точность определений групп. Запускает следующую команду для исправления и создания отчетов об ошибках: `% grpck -y ALL`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Да **Стандартные настройки AIX** Никакого действия	Нет
Обновление TCB	Использует команду **tcbck** для проверки и обновления TCB. Запускает следующую команду: `% tcbck -y ALL` **Примечание:** Если TCB требуется в системе, это правило не будет выполнено при выключенном TCB. Предварительно требуемое правило (prereqtcb) также не будет выполнено с предупреждением. **Предварительное требование**: Необходимо выбрать TCB при установке системы.	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Да **Стандартные настройки AIX** Да	Нет

Проверить определения файлов	Использует команду **sysck** для проверки и исправления файловой базы `/etc/objrepos/inventory`:
	`% sysck -i -f \`
	`/etc/security/sysck.cfg.rte`
		Высокий уровень защиты
		Да
		Средний уровень защиты
		Да
		Низкий уровень защиты
		Да
		Стандартные настройки AIX
		Никакого действия
			Нет

Таблица 21. Эксперт безопасности AIX - Определения паролей и система пользовательских групп (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отменить
Проверить определения паролей	Проверяет точность определений паролей. Запускает следующую команду для исправления и создания отчетов об ошибках: `% pwdck -y ALL`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Да **Стандартные настройки AIX** Никакого действия	Нет
Проверить определения пользователей	Проверяет точность определений пользователей. Запускает следующую команду для исправления и создания отчетов об ошибках: `% usrck -y ALL`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Да **Стандартные настройки AIX** Никакого действия	Нет

Группа рекомендаций стратегии входа в система Эксперта безопасности AIX

Эксперт безопасности AIX предоставляет особые настройки стратегии входа в систему.

Примечание: Для того, чтобы обеспечить отслеживаемость связанных с безопасностью задач, рекомендуется, чтобы пользователи входили в систему, используя обычный ИД пользователя, а потом запускали *команду* **su**, для того, чтобы выполнять команды, используя учетную запись root, а не входили в систему сразу, используя учетную запись root. В таком случае, система может позволить различным пользователям выполнять задачи с использованием учетной записи root, если несколько пользователей знают и используют пароль этой учетной записи.

Таблица 22. Эксперт безопасности AIX - Рекомендации стратегии входа в систему

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме-на
Промежуток времени для неудачных попыток войти в систему	Задает соответствующее значение для атрибута **logininterval** в `/etc/security/login.cfg`, который указывает промежуток времени (в секундах), в течение которого могут производиться неудачные попытки войти в систему, перед тем, как порт будет заблокирован. Например, если **logininterval** равен 60 и **logindisable** равен 4, то учетная запись будет заблокирована, если в течение одной минут произведено четыре неудачные попытки войти в систему.	**Высокий уровень защиты** 300 **Средний уровень защиты** 60 **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Без ограничений	Да

Таблица 22. Эксперт безопасности AIX - Рекомендации стратегии входа в систему (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отмена
Число попыток войти в систему пред блокирование учетной записи	Задает соответствующее значение для атрибута **loginretries** в /etc/security/user, который указывает число последовательных попыток входа в систему для учетной записи, перед тем, как она будет заблокирована. Не включать для учетной записи root.	**Высокий уровень защиты** 3 **Средний уровень защиты** 4 **Низкий уровень защиты** 5 **Стандартные настройки AIX** Без ограничений	Да
Удаленный доступ к учетной записи root	Изменяет значение атрибута **rlogin** в /etc/security/user, который указывает, разрешено ли использование удаленного доступа к системе для учетной записи root.	**Высокий уровень защиты** Ложь **Средний уровень защиты** Ложь **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Истина	Да
Разрешить вход в систему после блокировки	Задает соответствующее значение для атрибута **loginreenable** в /etc/security/login.cfg, который указывает промежуток времени (в секундах), по прошествии которого, будет разблокирован порт, заблокированный из-за параметра **logindisable**.	**Высокий уровень защиты** 360 **Средний уровень защиты** 30 **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Без ограничений	Да
Заблокировать вход в систему после неудачных попыток войти в систему	Задает соответствующее значение для атрибута **logindisable** в /etc/security/login.cfg, который указывает число неудачных попыток входа в систему, перед тем как порт будет заблокирован.	**Высокий уровень защиты** 10 **Средний уровень защиты** 10 **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Без ограничений	Да
Тайм-аут входа в систему	Задает соответствующее значение для атрибута **logintimeout** в /etc/security/login.cfg, который задает промежуток времени, в течение которого можно вводить пароль.	**Высокий уровень защиты** 30 **Средний уровень защиты** 60 **Низкий уровень защиты** 60 **Стандартные настройки AIX** 60	Да

Таблица 22. Эксперт безопасности AIX - Рекомендации стратегии входа в систему (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отмена
Промежуток времени между неудачными попытками войти в систему	Задает соответствующее значение для атрибута **logindelay** в /etc/security/login.cfg, который указывает промежуток времени (в секундах) между неудачными попытками войти в систему. Дополнительная задержка срабатывает после каждого неудачного входа в систему. Например, если для **logindelay** равен 5, то перед следующим запросом терминал будет ждать пять секунд после первой неудачной попытки войти в систему. После второй неудачной попытки войти в систему, терминал будет ждать 10 секунд (2*5), а после третьей неудачной попытки войти в систему, терминал будет ждать 15 секунд (3*5).	**Высокий уровень защиты** 10 **Средний уровень защиты** 4 **Низкий уровень защиты** 5 **Стандартные настройки AIX** Без ограничений	Да
Локальный вход в систему	Изменяет значение атрибута **login** в /etc/security/user, который указывает, разрешен ли вход в систему через консоль для учетной записи root.	**Высокий уровень защиты** Ложь **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Истина	Да

Группа рекомендаций стратегии контроля Эксперта безопасности AIX

Эксперт безопасности AIX предоставляет особые настройки стратегии контроля.

Как и для других настроек защиты, для контроля лотков необходимо, чтобы перед применением любых правил контроля для Высокого, Среднего или Низкого уровня защиты выполнялись правила анализа (предварительно). Для контроля лотков, необходимо, чтобы были выполнены следующие правила анализа:

1. Предварительное правило для контроля проверяет, что в настоящее время контроль не запущен. Если контроль запущен, значит настройки уже были заданы и Эксперт безопасности AIX не должен менять существующие параметры и процедуру контроля.
2. Необходимо не менее 100 мегабайт свободного пространства в группе томов, которая изменяется автоматически, или существующая файловая система /audit должна иметь размер не менее 100.

Если указанные выше обязательные условия выполнены и опции контроля выбраны в Эксперт безопасности AIX, то Эксперт безопасности AIX настроит и активирует контроль системы следующим образом. Кнопка Эксперт безопасности AIX **Включить контроль лотков** задает стратегию контроля. Контроль должен быть включен в системе.

1. Перед запуском контроля необходимо создать и смонтировать файловую систему JFS /audit. Размер файловой системы должен быть не менее 100 мегабайт.
2. Необходимо запустить контроль в режиме лотка. Файл /etc/security/audit/config необходимо настроить следующим образом:

```
start:
            binmode = on
            streammode = off
bin:
            trail = /audit/trail
            bin1 = /audit/bin1
            bin2 = /audit/bin2
            binsize = 10240
            cmds
```

```
= /etc/security/audit/bincmds
      .
      .
      etc
```

3. Добавьте контрольные записи корневого каталога и обычного пользователя для высокого, среднего и низкого уровней защиты.
4. Контроль необходимо включить при перезагрузке для высокого, среднего и низкого уровней защиты.
5. Контроль необходимо включить для новых пользователей для высокого, среднего и низкого уровней защиты. Это можно сделать, добавив запись `auditclasses` в пользовательский раздел в файл `/usr/lib/security/mkuser.default`.
6. Во избежание переполнения файловой системы `/audit`, необходимо добавить **cronjob**.

Правило отмены контроля выключает контроль и удаляет включение контроля при перезагрузке.

В следующих таблицах приведены списки значений, которые задает Эксперт безопасности AIX для **включения контроля лотков**:

Таблица 23. Значения, которые задает Эксперт безопасности AIX для включения контроля лотков

Защита высокого уровня	Защита среднего уровня	Защита низкого уровня	Стандартные настройки AIX

Добавить следующие записи контроля для корневого каталога и обычного пользователя:
`Root:`
`General`
`Src`
`Mail`
`Cron`
`Tcpip`
`Ipsec`
`Lvm`
`User:`
`General`
`Src`
`Cron`
`Tcpip`
 	
Добавьте следующую запись в пользовательский раздел файла `/usr/lib/security/mkuser.default` для включения контроля для создаваемых пользователей:
`auditclasses=general,SRC,\`
`cron,tcpip`
	Добавить следующие записи контроля для корневого каталога и обычного пользователя:
	`Root:`
	`General`
	`Src`
	`Tcpip`
	`User:`
	`General`
	`Tcpip`
	
	Добавьте следующую запись в пользовательский раздел файла `/usr/lib/security/mkuser.default` для включения контроля для создаваемых пользователей:
	`auditclasses=general,`
	`tcpip`
		Добавить следующие записи контроля для корневого каталога и обычного пользователя:
		`Root:`
		`General`
		`Tcpip`
		`User:`
		`General`
		
		Добавьте следующую запись в пользовательский раздел файла `/usr/lib/security/mkuser.default` для включения контроля для создаваемых пользователей:
		`auditclasses=general`
			Файл /etc/security/audit/config содержит следующую запись:
			`default=login`
			
			Идентификации класса контроля определяется следующим образом:
			`login = USER_SU,`
			`USER_Login,`
			`USER_Logout,`
			`TERM_Logout,`
			`USER_Exit`
			
			Примечание: Компонент стандартных параметров выключает контроль.

Таблица 23. Значения, которые задает Эксперт безопасности AIX для включения контроля лотков (продолжение)

Защита высокого уровня	Защита среднего уровня	Защита низкого уровня	Стандартные настройки AIX
Добавить следующие записи контроля для корневого каталога и обычного пользователя: **root:** general src mail cron tcpip ipsec lvm aixpert *Пользователь:* general src cron tcpip Добавьте следующую запись в пользовательский раздел файла `/usr/lib/security/mkuser.default` для включения контроля для создаваемых пользователей: `auditclasses=general,SRC, cron,tcpip`	Добавить следующие записи контроля для корневого каталога и обычного пользователя: **root:** general src tcpip aixpert *Пользователь:* general tcpip Добавьте следующую запись в пользовательский раздел файла `/usr/lib/security/mkuser.default` для включения контроля для создаваемых пользователей: `auditclasses=general, tcpip`	Добавить следующие записи контроля для корневого каталога и обычного пользователя: **root:** general tcpip aixpert *Пользователь:* general Добавьте следующую запись в пользовательский раздел файла `/usr/lib/security/mkuser.default` для включения контроля для создаваемых пользователей: `auditclasses=general`	Да

Необходимо, чтобы каждый час запускался cronjob для проверки размера `/audit`. Если параметр Соотношение свободного пространства контроля равен true, необходимо выполнить действия по копированию журнала контроля. Параметр Соотношение свободного пространства контроля определяет, что файловая система `/audit` не переполнена. Если файловая система `/audit` переполнена, выполняются действия по копированию журнала контроля (отключение контроля, сохранение резервной копии `/audit/trail` в `/audit/trailOneLevelBack` и включение контроля).

Эксперт безопасности AIX группа записей /etc/inittab

Эксперт безопасности AIX добавляет символы комментария для определенных записей в `/etc/inittab`, в результате они не запускаются при загрузке системы.

Таблица 24. Эксперт безопасности AIX - Записи /etc/inittab

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отмена
Отключить **qdaemon** / Включить **qdaemon**	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inittab: `qdaemon:2:wait:/usr/bin/startsrc -sqdaemon`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Комментарий **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Удалить символы комментария	Да
Отключить демон **lpd** / Включить демон **lpd**	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inittab: `lpd:2:once:/usr/bin/startsrc -s lpd`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Комментарий **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Удалить символы комментария	Да
Отключить CDE / Включить CDE	Если в системе не настроен LFT, добавляет или удаляет символы комментария для следующей записи в /etc/inittab: `dt:2:wait:/etc/rc.dt`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Комментарий **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Удалить символы комментария	Да
Отключить демон **piobe** / Включить демон **piobe**	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inittab: `piobe:2:wait:/usr/lib/lpd/pio/etc/pioinit >/dev/null 2>&1`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Комментарий **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Удалить символы комментария	Да

Группа рекомендаций настройки /etc/rc.tcpip Эксперта безопасности AIX

Эксперт безопасности AIX добавляет символы комментария для определенных записей в `/etc/rc.tcpip`, в результате они не запускаются при загрузке системы.

В следующих таблицах приведены списки записей, которые взяты в символы комментария в `/etc/rc.tcpip`, в результате они не запускаются при загрузке системы.

Таблица 25. Настройки Эксперт безопасности AIX `/etc/rc.tcpip`

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отмена
Отключить почтовый клиент / Включить почтовый клиент	Добавляет символы комментария или удаляет символы комментария для следующей записи в `/etc/rc.tcpip`: `start /usr/lib/sendmail "$src_running"`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Удалить символы комментария	Да
Отключить демон маршрутизации	Добавляет символы комментария для следующей записи в `/etc/rc.tcpip`: `start /usr/sbin/routed "$src_running" -q`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить демон **mrouted**	Добавляет символы комментария для следующей записи в `/etc/rc.tcpip`: `start /usr/sbin/mrouted "$src_running"`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить демон **timed**	Добавляет символы комментария для следующей записи в `/etc/rc.tcpip`: `start /usr/sbin/timed`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Да **Стандартные настройки AIX** Да	Да

Таблица 25. Настройки Эксперт безопасности AIX /etc/rc.tcpip (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отмена
Отключить демон **rwhod**	Добавляет символы комментария для следующей записи в /etc/rc.tcpip: `start /usr/sbin/rwhod "$src_running"`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить демон печати	Добавляет символы комментария для следующей записи в /etc/rc.tcpip: `start /usr/sbin/lpd "$src_running"`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить демон SNMP / Включить демон SNMP	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/rc.tcpip: `start /usr/sbin/snmpd "$src_running"`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Комментарий **Низкий уровень защиты** Выключает демон SNMP **Стандартные настройки AIX** Удалить символы комментария	Да
Остановить DHCP Agent	Добавляет символы комментария для следующей записи в /etc/rc.tcpip: `start /usr/sbin/dhcprd "$src_running"`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Остановить сервер DHCP	Добавляет символы комментария для следующей записи в /etc/rc.tcpip: `start /usr/sbin/dhcpsd "$src_running"`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да

Таблица 25. Настройки Эксперт безопасности AIX /etc/rc.tcpip (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отмена
Остановить autoconf6	Добавляет символы комментария для следующей записи в /etc/rc.tcpip: `start /usr/sbin/autoconf6 ""`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Остановить демон DNS	Добавляет символы комментария для следующей записи в /etc/rc.tcpip: `start /usr/sbin/named "$src_running"`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить демон **gated**	Добавляет символы комментария для следующей записи в /etc/rc.tcpip: `start /usr/sbin/gated "$src_running"`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Да **Стандартные настройки AIX** Да	Да
Остановить DHCP Client	Добавляет символы комментария для следующей записи в /etc/rc.tcpip: `start /usr/sbin/dhcpd "$src_running"`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить демон DPID2	Добавляет символы комментария для следующей записи в /etc/rc.tcpip: `start /usr/sbin/dpid2 "$src_running"`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да

Таблица 25. Настройки Эксперт безопасности AIX /etc/rc.tcpip (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отмена
Остановить демон NTP	Добавляет символы комментария для следующей записи в /etc/rc.tcpip: `start /usr/sbin/xntpd "$src_running"`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да

Группа рекомендаций настройки /etc/inetd.conf Эксперта безопасности AIX

Эксперт безопасности AIX снабжает комментариями особые записи в /etc/inetd.conf.

Установка AIX по умолчанию запускает несколько сетевых служб, которые могут конфликтовать со службой безопасности системы. Эксперт безопасности AIX отключает ненужные и небезопасные службы путем добавления символов комментария для соответствующих записей в файле /etc/inetd.conf. При использования стандартных настроек AIX эти записи не взяты в символы комментария. В следующих таблицах приведены списки записей, которые взяты или не взяты в символы комментария в /etc/inetd.conf.

Таблица 26. Эксперт безопасности AIX - Настройки /etc/inetd.conf

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
Отключить **sprayd** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `sprayd sunrpc_udp udp wait root \ /usr/lib/netsvc/`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить службу UDP chargen в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `chargen dgram udp wait root internal`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да

Таблица 26. Эксперт безопасности AIX - Настройки /etc/inetd.conf (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на

Отключить telnet / Включить telnet	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inetd.conf:
	`telnet stream tcp6 nowait root \`
	`/usr/sbin/telnetd telnetd`
		Высокий уровень защиты
		Комментарий
		Средний уровень защиты
		Никакого действия
		Низкий уровень защиты
		Никакого действия
		Стандартные настройки AIX
		Удалить символы комментария
			Да

Отключить службу UDP Echo в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `echo dgram udp wait root internal`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да

Отключить **tftp** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf:
	`tftp dgram udp6 SRC nobody \`
	`/usr/sbin/tftpd tftpd -n`
		Высокий уровень защиты
		Да
		Средний уровень защиты
		Да
		Низкий уровень защиты
		Никакого действия
		Стандартные настройки AIX
		Да
			Да

Отключить демон **krshd**	Добавляет символы комментария для следующей записи в /etc/inetd.conf:
	`kshell stream tcp nowait root \`
	`/usr/sbin/krshd krshd`
		Высокий уровень защиты
		Да
		Средний уровень защиты
		Никакого действия
		Низкий уровень защиты
		Никакого действия
		Стандартные настройки AIX
		Да
			Да

Отключить **rusersd** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf:
	`rusersd sunrpc_udp udp wait root \`
	`/usr/lib/netsvc/`
		Высокий уровень защиты
		Да
		Средний уровень защиты
		Да
		Низкий уровень защиты
		Никакого действия
		Стандартные настройки AIX
		Да
			Да

Таблица 26. Эксперт безопасности AIX - Настройки /etc/inetd.conf (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
Отключить **rexecd** в /etc/inetd.conf / Включить **rexecd** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `exec stream tcp6 nowait root \ /usr/sbin/rexecd rexecd`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Комментарий **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Удалить символы комментария	Да
Отключить POP3D	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `pop3 stream tcp nowait root \ /usr/sbin/pop3d pop3d`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить **pcnfsd** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `pcnfsd sunrpc_udp udp wait root \ /usr/sbin/rpc.pcnfsd pcnfsd`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить **bootpd** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `bootps dgram udp wait root \ /usr/sbin/bootpd`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить **rwalld** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `rwalld sunrpc_udp udp wait root \ /usr/lib/netsvc/`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да

Таблица 26. Эксперт безопасности AIX - Настройки /etc/inetd.conf (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
Отключить службу UDP discard в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `discard dgram udp wait root \ internal`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить службу TCP daytime в /etc/inetd.conf / Включить службу TCP daytime в /etc/inetd.conf	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inetd.conf: `daytime stream tcp nowait root \ internal`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Удалить символы комментария	Да
Отключить **netstat** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `netstat stream tcp nowait nobody \ /usr/bin/netstat`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить демон **rshd** / Включить демон **rshd**	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inetd.conf: `shell stream tcp6 nowait root \ /usr/sbin/rshd rshd rshd`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Комментарий **Низкий уровень защиты** Комментарий **Стандартные настройки AIX** Удалить символы комментария	Да
Отключить службу **cmsd** в /etc/inetd.conf / Включить службу **cmsd** в /etc/inetd.conf	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inetd.conf: `cmsd sunrpc_udp udp wait root \ /usr/dt/bin/rpc.cms cmsd`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Удалить символы комментария	Да

Таблица 26. Эксперт безопасности AIX - Настройки /etc/inetd.conf (продолжение)

Имя кнопки	Описание	Значение, заданное Экспертом безопасности AIX	Отме- на

Отключить службу **ttdbserver** в /etc/inetd.conf / Включить службу **ttdbserver** в /etc/inetd.conf	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inetd.conf:
	`ttdbserver sunrpc_tcp tcp wait \`
	`root /usr/dt/bin/`
		Высокий уровень защиты
		Комментарий
		Средний уровень защиты
		Никакого действия
		Низкий уровень защиты
		Никакого действия
		Стандартные настройки AIX
		Удалить символы комментария
			Да

Отключить **uucpd** в /etc/inetd.conf / Включить **uucpd** в /etc/inetd.conf	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inetd.conf:
	`uucp stream tcp nowait root \`
	`/usr/sbin/uucpd uucpd`
		Высокий уровень защиты
		Комментарий
		Средний уровень защиты
		Никакого действия
		Низкий уровень защиты
		Никакого действия
		Стандартные настройки AIX
		Удалить символы комментария
			Да

Отключить службу UDP time в /etc/inetd.conf / Включить службу UDP time в /etc/inetd.conf	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inetd.conf: `time dgram udp wait root internal`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Удалить символы комментария	Да

Отключить службу TCP time в /etc/inetd.conf / Включить службу TCP time в /etc/inetd.conf	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inetd.conf:
	`time stream tcp nowait root \`
	`internal`
		Высокий уровень защиты
		Комментарий
		Средний уровень защиты
		Никакого действия
		Низкий уровень защиты
		Никакого действия
		Стандартные настройки AIX
		Удалить символы комментария
			Да

Отключить **rexd** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf:
	`rexd sunrpc_tcp tcp wait root \`
	`/usr/sbin/tpc.rexd.rexd rexd`
		Высокий уровень защиты
		Да
		Средний уровень защиты
		Да
		Низкий уровень защиты
		Да
		Стандартные настройки AIX
		Да
			Да

Таблица 26. Эксперт безопасности AIX - Настройки /etc/inetd.conf (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
Отключить службу TCP chargen в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `chargen stream tcp nowait root \ internal`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить **rlogin** в /etc/inetd.conf / Включить **rlogin** в /etc/inetd.conf	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inetd.conf: `login stream tcp6 nowait root \ /usr/sbin/rlogind rlogind`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Комментарий **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Удалить символы комментария	Да
Отключить службу talk в /etc/inetd.conf	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inetd.conf: `talk dgram udp wait root \ /usr/sbin/talkd talkd`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Комментарий **Низкий уровень защиты** Комментарий **Стандартные настройки AIX** Удалить символы комментария	Да
Отключить **fingerd** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `finger stream tcp nowait nobody \ /usr/sbin/fingerd fingerd`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить FTP / Включить FTP	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inetd.conf: `ftp stream tcp6 nowait root \ /usr/sbin/ftpd ftpd`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Удалить символы комментария	Да

Таблица 26. Эксперт безопасности AIX - Настройки /etc/inetd.conf (продолжение)

Имя кнопки	Описание	Значение, заданное Экспертом безопасности AIX	Отме- на

Отключить IMAPD	Добавляет символы комментария для следующей записи в /etc/inetd.conf:
	`imap2 stream tcp nowait root \`
	`/usr/sbin/imapd imapd`
		Высокий уровень защиты
		Да
		Средний уровень защиты
		Никакого действия
		Низкий уровень защиты
		Никакого действия
		Стандартные настройки AIX
		Да
			Да

Отключить **comsat** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf:
	`comsat dgram udp wait root \`
	`/usr/sbin/comsat comsat`
		Высокий уровень защиты
		Да
		Средний уровень защиты
		Никакого действия
		Низкий уровень защиты
		Никакого действия
		Стандартные настройки AIX
		Да
			Да

Отключить **rquotad** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf:
	`rquotad sunrpc_udp udp wait root \`
	`/usr/sbin/rpc.rquotad`
		Высокий уровень защиты
		Да
		Средний уровень защиты
		Да
		Низкий уровень защиты
		Да
		Стандартные настройки AIX
		Да
			Да

Отключить службу UDP daytime в /etc/inetd.conf / Включить службу UDP daytime в /etc/inetd.conf	Добавляет символы комментария или удаляет символы комментария для следующей записи в /etc/inetd.conf: `daytime dgram udp wait root internal`	**Высокий уровень защиты** Комментарий **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Удалить символы комментария	Да

Отключить **krlogind** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf:
	`klogin stream tcp nowait root \`
	`/usr/sbin/krlogind krlogind`
		Высокий уровень защиты
		Да
		Средний уровень защиты
		Никакого действия
		Низкий уровень защиты
		Никакого действия
		Стандартные настройки AIX
		Да
			Да

Таблица 26. Эксперт безопасности AIX - Настройки /etc/inetd.conf (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
Отключить службу TCP Discard в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `discard stream tcp nowait root \ internal`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить службу TCP echo в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `echo stream tcp nowait root internal`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить **sysstat** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `systat stream tcp nowait nodby \ /usr/bin/ps ps -ef`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить **rstatd** в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `rstatd sunrpc_udp udp wait root \ /usr/sbin/rpc.rstatd rstatd`	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить службу dtspc в /etc/inetd.conf	Добавляет символы комментария для следующей записи в /etc/inetd.conf: `dtspc stream tcp nowait root \ /usr/dt/bin/dtspcd`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да

Группа отключения SUID для команд AIX

По умолчанию, следующие команды установлены с битовым набором SUID. Для высокого, среднего и низкого уровня безопасности этот битовый набор не задан. Для стандартных настроек AIX битовый набор SUID восстановлен для этих команд.

Таблица 27. Отключение SUID в командах - Эксперт безопасности AIX

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
hls_filepermgr	Администратор прав доступа к файлам: Запускает команду **fpm** с опцией high для удаления setuid и setgid из команд, имеющих права доступа	Защита высокого уровня	Да
mls_filepermgr	Администратор прав доступа к файлам: Запускает команду **fpm** с опцией medium для удаления setuid и setgid из команд, имеющих права доступа	Защита среднего уровня	Да
lls_filepermgr	Администратор прав доступа к файлам: Запускает команду **fpm** с опцией low для удаления setuid и setgid из команд, имеющих права доступа	Защита низкого уровня	Да

Группа блокирования удаленных служб Эксперта безопасности AIX

Эксперт безопасности AIX блокирует небезопасные команды при высоком и среднем уровне защиты.

Следующие команды и демоны часто используются для того, чтобы отыскать брешь в системе безопасности. Если установлен высокий или средний уровень безопасности, эти небезопасные команды не получают права доступа для выполнения, а демоны блокируются. Если установлен низкий уровень безопасности, в отношении этих команд и демонов действий не предпринимается. Если установлены Стандартные настройки AIX, данные команды и демоны допускаются к использованию.

- **rcp**
- **rlogin**
- **rsh**
- **tftp**
- **rlogind**
- **rshd**
- **tftpd**

Таблица 28. Эксперт безопасности AIX - Блокирование удаленных служб

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
Включить использование небезопасных демонов	Если включена TCB, разрешает запуск демонов **rlogind**, **rshd** и **tftpd**, добавляет в базу данных sysck изменения режима для этих демонов. Если TCB не включена, разрешается запуск демонов **rlogind**, **rshd** и **tftpd**.	**Высокий уровень защиты** Никакого действия **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Никакого действия	Да
Отключить небезопасные команды	1. Если включена TCB, блокирует выполнение команд **rcp**, **rlogin**, **rsh** и **tftp** и добавляет в базу данных **sysck** изменения режима для этих команд. Если TCB не включена, блокирует выполнение команд **rcp**, **rlogin** и **rsh**. 2. Останавливает текущие экземпляры команд **rcp**, **rlogin**, **rsh**, **tftp** и **uftp**, пока какая-либо из этих команд не окажется родительским процессом для Эксперт безопасности AIX. 3. Добавляет раздел **tcpip:** в /etc/security/config, чтобы ограничить использование **.netrc** для **ftp** и **rexec**.	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Никакого действия	Да

Таблица 28. Эксперт безопасности AIX - Блокирование удаленных служб (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме-на
Включить использование небезопасных команд	1. Если TCB включена, разрешает запуск команд **rcp**, **rlogin**, **rsh** и **tftp** и добавляет в базу данных **sysck** изменения режима для этих команд. Если TCB не включена, разрешается запуск команд **rcp**, **rlogin** и **rsh**. 2. Удаляет файл `/etc/security/config`.	**Высокий уровень защиты** Никакого действия **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да
Отключить использование небезопасных демонов	1. Если включена TCB, блокирует доступ для демонов **rlogind**, **rshd** и **tftpd** к системе и добавляет в базу данных **sysck** изменения режима для этих демонов. Если TCB не включена, блокирует доступ демонов **rlogind**, **rshd** и **tftpd** к системе. 2. Останавливает текущие экземпляры демонов **rlogind**, **rshd** и **tftpd**, пока какой-либо из этих демонов не окажется родительским процессом для Эксперт безопасности AIX.	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Никакого действия	Да
Остановить демон NFS	• Удалить все точки монтирования NFS • Отключает NFS • Удаляет сценарий запуска NFS из `/etc/inittab`	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Никакого действия	Да
Включить демон NFS	• Экспортирует все записи из `/etc/exports` • Добавляет запись в /etc/inittab для запуска `/etc/rc.nfs` при перезапуске системы • Сразу же запускает `/etc/rc.nfs`	**Высокий уровень защиты** Никакого действия **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Да	Да

Группа удаленного доступа, при котором не требуется идентификация, Эксперта безопасности AIX

AIX поддерживает несколько служб, для которых не требуется идентификации пользователя для входа в систему.

Файл `/etc/hosts.equiv`, любые локальные хосты определения файлов **`$HOME`**`/.rhosts` и учетные записи пользователей, которые могут запускать удаленные команды на локальном хосте без пароля. При необходимости иметь эту функцию, эти файлы должны быть удалены.

Таблица 29. Эксперт безопасности AIX - Удаленный доступ, при котором не требуется идентификация

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
Удаленные службы rhosts и netrc	В файлах `.rhosts` и `.netrc` хранятся имена и пароли пользователей в текстовом формате, удобном для обработки.	**Высокий уровень защиты** Удалить файлы `.rhosts` и `.netrc` из домашних каталогов всех пользователей, включая учетную запись root. **Средний уровень защиты** Удалить файлы `.rhosts` и `.netrc` из домашних каталогов всех пользователей, включая учетную запись root. **Низкий уровень защиты** Удалить файлы `.rhosts` и `.netrc` из домашнего каталога учетной записи root. **Стандартные настройки AIX** Удалить файлы `.rhosts` и `.netrc` из домашних каталогов всех пользователей, включая учетную запись root.	Да
Удалить записи из файла `/etc/hosts.equiv`	Файл `/etc/hosts.equiv` как и файл локального пользователя **`$HOME`**`/.rhosts`, определяет, каким внешним пользователям разрешено удаленно запускать команды на локальном хосте. Если пользователю внешнего хоста станут известны имя и пользователя и хоста, он сможет запускать удаленные команды на локальном хосте без идентификации.	**Высокий уровень защиты** Удалить все записи из `/etc/hosts.equiv`. **Средний уровень защиты** Удалить все записи из `/etc/hosts.equiv`. **Низкий уровень защиты** Удалить все записи из `/etc/hosts.equiv`. **Стандартные настройки AIX** Удалить все записи из `/etc/hosts.equiv`.	Да

Группа настройки опций сети Эксперта безопасности AIX

Настойка опций сети является значительной частью настройки системы безопасности. Если значение атрибута сети равно 0, опция отключена, если значение атрибута сети равно 1, опция включена.

В следующей таблице приведен список настроек атрибутов сети для высокого, среднего и низкого уровней защиты. Также в этой таблице приведены описано, как отдельные опции сети связаны с обеспечением безопасности сети.

Таблица 30. Эксперт безопасности AIX - Настройка опций сети для обеспечения безопасности сети

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
Опция сети ipsrcrouteforward	Указывает, будет ли система пересылать пакеты, отправляемые по сложным маршрутам ICMP. Отключив ipsrcrouteforward, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.	**Защита высокого уровня** 0 **Защита среднего уровня** 0 **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** 1	Да

Таблица 30. Эксперт безопасности AIX - Настройка опций сети для обеспечения безопасности сети (продолжение)

Имя кнопки	Описание	Значение, заданное Экспертом безопасности AIX	Отмена
Опция сети ipignoreredirects	Указывает будет ли система обрабатывать запросы на перенаправление пакетов.	**Защита высокого уровня** 1 **Защита среднего уровня** Никакого действия **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** Без ограничений	Да
Опция сети clean_partial_conns	Указывает, будет ли система избегать атак синхронизации (SYN).	**Защита высокого уровня** 1 **Защита среднего уровня** 1 **Защита низкого уровня** 1 **Стандартные настройки AIX** Без ограничений	Да
Опция сети ipsrcrouterecv	Указывает, будет ли система получать пакеты, отправляемые по сложным маршрутам ICMP. Отключив ipsrcrouterecv, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.	**Защита высокого уровня** 0 **Защита среднего уровня** Никакого действия **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** Без ограничений	Да
Опция сети ipforwarding	Указывает, будет ли ядро выполнять пересылку пакетов. Отключив ipforwarding, можно запретить перенаправление пакетов в другие сети.	**Защита высокого уровня** 0 **Защита среднего уровня** Никакого действия **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** Без ограничений	Да
Опция сети ipsendredirects	Указывает, будет ли ядро передавать сигналы о перенаправлении пакетов. Отключив ipsendredirects, можно запретить перенаправление пакетов в другие сети.	**Защита высокого уровня** 0 **Защита среднего уровня** Никакого действия **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** 1	Да

Таблица 30. Эксперт безопасности AIX - Настройка опций сети для обеспечения безопасности сети (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме-на
Опция сети ip6srcrouteforward	Указывает, будет ли система пересылать пакеты IPv6, отправляемые по сложным маршрутам ICMP. Отключив ip6srcrouteforward, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.	**Защита высокого уровня** 0 **Защита среднего уровня** Никакого действия **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** 1	Да
Опция сети ip6srcrouteforward	Указывает, разрешена ли прямая отправка широковещательных пакетов через шлюз. Отключив directed_broadcast, можно запретить перенаправление пакетов в удаленную сеть.	**Защита высокого уровня** 0 **Защита среднего уровня** 0 **Защита низкого уровня** 0 **Стандартные настройки AIX** Без ограничений	Да
Опция сети tcp_pmtu_discover	Включает или выключает режим определения MTU для приложений протокола TCP. Отключив tcp_pmtu_discover, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.	**Защита высокого уровня** 0 **Защита среднего уровня** 0 **Защита низкого уровня** 0 **Стандартные настройки AIX** 1	Да
Опция сети bcastping	Запрещает отвечать на эхозапросы ICMP, передаваемые в широковещательном режиме. Отключив bcastping, можно защититься от возможных атак типа smurf (атака, направленная на отказ в обслуживании путем отправки большого количества широковещательных запросов на конкретный IP-адрес).	**Защита высокого уровня** 0 **Защита среднего уровня** 0 **Защита низкого уровня** 0 **Стандартные настройки AIX** Без ограничений	Да
Опция сети icmpaddressmas	Указывает, будет ли система отвечать на запросы маски подсети по протоколу ICMP. Отключив icmpaddressmask, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.	**Защита высокого уровня** 0 **Защита среднего уровня** 0 **Защита низкого уровня** 0 **Стандартные настройки AIX** Без ограничений	Да

Таблица 30. Эксперт безопасности AIX - Настройка опций сети для обеспечения безопасности сети (продолжение)

Имя кнопки	Описание	Значение, заданное Экспертом безопасности AIX	Отмена
Опция сети udp_pmtu_discover	Включает или отключает вычисление MTU маршрута для приложений UDP. Отключив udp_pmtu_discover, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.	**Защита высокого уровня** 0 **Защита среднего уровня** 0 **Защита низкого уровня** 0 **Стандартные настройки AIX** 1	Да
Опция сети ipsrcroutesend	Указывает, разрешено ли приложениям отправлять пакеты по сложным маршрутам ICMP. Отключив ipsrcroutesend, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.	**Защита высокого уровня** 0 **Защита среднего уровня** Никакого действия **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** 1	Да
Опция сети nonlocsrcroute	Указывает, разрешена ли отправка пакетов IP по сложным маршрутам за пределы локальной сети. Отключив nonlocsrcroute, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.	**Защита высокого уровня** 0 **Защита среднего уровня** Никакого действия **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** Без ограничений	Да
Опция сети tcp_tcpsecure	Защищает уязвимые места соединения TCP. Значения: • 0 = защита отсутствует • 1 = отправка ложной команды SYN установленному соединению • 2 = отправка ложной команды RST установленному соединению • 3 = внедрение данных в установленное соединение TCP • 5-7 = комбинации указанных выше уязвимостей	**Защита высокого уровня** 7 **Защита среднего уровня** 7 **Защита низкого уровня** 5 **Стандартные настройки AIX** Без ограничений	Да
Опция сети sockthresh	Задает ограничение на использование сетевой памяти. Число сокетов не может превышать значение, указанное в параметре sockthresh. Указывает максимальный объем сетевой памяти, который может быть выделен сокетам.	**Защита высокого уровня** 60 **Защита среднего уровня** 70 **Защита низкого уровня** 85 **Стандартные настройки AIX** Без ограничений	Да

Следующие опции сети больше относятся к работе сети, чем к системе безопасности сети.

Таблица 31. Эксперт безопасности AIX - Настройка опций для работы сети

Имя кнопки	Описание	Значение, заданное Экспертом безопасности AIX	Отмена
Опция сети rfc1323	Опция rfc1323 позволяет точно настроить опцию масштабирования окна TCP.	**Защита высокого уровня** 1 **Защита среднего уровня** 1 **Защита низкого уровня** 1 **Стандартные настройки AIX** Без ограничений	Да
Опция сети tcp_sendspace	С помощью опции tcp_sendspace можно настроить количество данных, которые приложение может поместить в буфер ядра, перед тем, как оно будет заблокировано вызовом отправления.	**Защита высокого уровня** 262144 **Защита среднего уровня** 262144 **Защита низкого уровня** 262144 **Стандартные настройки AIX** 16384	Да
Опция сети tcp_mssdflt	Размер максимального сегмента по умолчанию, который используется для работы с удаленными сетями.	**Защита высокого уровня** 1448 **Защита среднего уровня** 1448 **Защита низкого уровня** 1448 **Стандартные настройки AIX** 1460	Да
Опция сети extendednetstats	Включает более подробную статистику для служб памяти сети.	**Защита высокого уровня** 1 **Защита среднего уровня** 1 **Защита низкого уровня** 1 **Стандартные настройки AIX** Без ограничений	Да
Опция сети tcp_recvspace	С помощью опции tcp_recvspace можно настроить количество байтов, которые принимающая система может поместить в буфер ядра в очереди приема сокетов.	**Защита высокого уровня** 262144 **Защита среднего уровня** 262144 **Защита низкого уровня** 262144 **Стандартные настройки AIX** 16384	Да

Таблица 31. Эксперт безопасности AIX - Настройка опций для работы сети (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
Опция сети sb_max	С помощью опции sb_max можно задать верхний предел количества буферов сокетов, помещенных в очередь отдельного сокета, который контролирует пространство использованное буферами, которые помещены в очередь отправителя или получателя сокета.	**Защита высокого уровня** 1048576 **Защита среднего уровня** 1048576 **Защита низкого уровня** 1048576 **Стандартные настройки AIX** 1048576	Да

Группа правил фильтров IPsec Эксперта безопасности AIX

Эксперт безопасности AIX предоставляет следующие фильтры IPsec.

Таблица 32. Эксперт безопасности AIX - Правила фильтров IPsec

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
Блокирование хоста на 5 минут	Блокируются пакеты, предназначенные для некоторых портов tcp и udp на хосте с известной уязвимостью. В течение пяти минут хост не будет принимать пакеты, предназначенные для этих портов.	**Высокий уровень защиты** Да **Средний уровень защиты** Никакого действия **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Никакого действия	Да
Защитить хост от сканирования портов	Защищает хост от сканирования портов. Любой удаленный хост, производящий сканирование портов блокируется на пять минут. В течение пяти минут не принимаются пакеты от данного удаленного хоста.	**Высокий уровень защиты** Да **Средний уровень защиты** Да **Низкий уровень защиты** Никакого действия **Стандартные настройки AIX** Никакого действия	Да

Группа Прочие Эксперта безопасности AIX

Эксперт безопасности AIX предоставляет прочие настройки безопасности для высокого, среднего и низкого уровней защиты.

Таблица 33. Прочие рекоменадции - Эксперт безопасности AIX

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
Удалите точку в пути root	Проверяет файлы **$HOME/.profile**, **$HOME/.kshrc**, **$HOME/.cshrc** и **$HOME/.login** на наличие точек (.) в переменной среды PATH и удаляет их, если они существуют. **Примечание:** Точки удаляются только если запись в файле начинается с переменной среды PATH и содержит точки (.). Файл не изменяется, если переменная среды PATH содержит другие переменные или ей присвоено значение, возвращенное программой, которая вызвана из сценария. Ниже приведен пример пути, который не будет изменен, где *pathprog* - это программа, которая возвращает строку пути: `PATH="$(pathprog)"` В этом пути точки удаляются из пути перед определением значения переменной *pathprog*, поэтому из возвращаемого пути точки не удаляются.	**High Level Security** Да **Защита среднего уровня** Да **Защита низкого уровня** Да **Стандартные настройки AIX** Да	Да
Ограничить доступ к системе	Проверяет, что задачи **cron** можно запускать только через учетную запись root.	**High Level Security** Делает файл `cron.allow` доступным только для учетной записи root и удаляет файл `cron.deny`. **Защита среднего уровня** Никакого действия **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** Удаляет файл `cron.allow` и удаляет все записи в файле `cron.deny`.	Да
Удалить точку в `/etc/environment`	Удаляет точки (.) из переменной среды **PATH** в файле `/etc/environment`.	**High Level Security** Да **Защита среднего уровня** Да **Защита низкого уровня** Да **Стандартные настройки AIX** Да	Да
Удалить точку в пути не-root	Удаляет точки (.) в переменной среды **PATH** в файлах **$HOME/.profile**, **$HOME/.kshrc**, **$HOME/.cshrc** и **$HOME/.login** всех учетных записей, кроме root. **Примечание:** Точки удаляются только если запись в файле начинается с переменной среды PATH и содержит точки (.). Файл не изменяется, если переменная среды PATH содержит другие переменные или ей присвоено значение, возвращенное программой, которая вызвана из сценария. Ниже приведен пример пути, который не будет изменен, где *pathprog* - это программа, которая возвращает строку пути: `PATH="$(pathprog)"` В этом пути точки удаляются из пути перед определением значения переменной *pathprog*, поэтому из возвращаемого пути точки не удаляются.	**High Level Security** Да **Защита среднего уровня** Никакого действия **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** Никакого действия	Нет

Таблица 33. Прочие рекоменадции - Эксперт безопасности AIX (продолжение)

Имя кнопки	Описание	Значение, заданное Экспert безопасности AIX	Отме- на
Добавить пользователя root в файле /etc/ftpusers	Добавить имя пользователя root в файл /etc/ftpusers, для того, чтобы запретить удаленный **ftp** через учетную запись root.	**High Level Security** Да **Защита среднего уровня** Да **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** Да	Да
Удалить пользователя root из файла /etc/ftpusers	Удаляет запись root из /etc/ftpusers, для того, чтобы разрешить удаленный доступ **ftp** с помощью учетной записи root.	**High Level Security** Никакого действия **Защита среднего уровня** Никакого действия **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** Да	Да

Создать пользователя herald	Проверяет, что в /etc/security/login.cfg отсутствует значение herald. Если значение по умолчанию herald занято, необходимо его изменить. Значение herald можно изменить только если локалью системы назначена локаль **en_US** или другая английская локаль. Если эти требования удовлетворены, значение атрибута herald в разделе по умолчанию в файле /etc/security/login.cfg является следующим:
	`Unauthorized use of this \`
	`system is prohibited.\nlogin:`
	
	Примечание: Настройки безопасности вступают в силу только для новых сеансов. Настройки безопасности не вступают в силу в тех сеансах, в которых были заданы эти настройки.
		High Level Security
		herald="Unauthorized use of this system is prohibited.\nlogin:"
		Защита среднего уровня
		herald="Unauthorized use of this system is prohibited.\nlogin:"
		Защита низкого уровня
		herald="Unauthorized use of this system is prohibited.\nlogin:"
		Стандартные настройки AIX
		herald=
			Да

Удалить учетную запись guest	Удаляет учетную запись guest и все данные этой учетной записи для высокого, среднего или низкого уровня безопасности. При использовании стандартных настроек AIX, учетная запись guest создается в системе. **Примечание:** Для этой учетной записи пароль должен задавать администратор напрямую, поскольку Эксперт безопасности AIX не выполняет подобные интерактивные задачи.	**High Level Security** Удалить данные и учетную запись guest **Защита среднего уровня** Удалить данные и учетную запись guest **Защита низкого уровня** Удалить данные и учетную запись guest **Стандартные настройки AIX** Добавить учетную запись guest.	Да

Таблица 33. Прочие рекоменадции - Эксперт безопасности AIX (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
Права доступа Crontab	Проверяет, что задачи **crontab** учетной записи root принадлежат и доступны для записи только для учетной записи root.	**High Level Security** Да **Защита среднего уровня** Да **Защита низкого уровня** Да **Стандартные настройки AIX** Никакого действия	Да
Разрешить доступ X-Server	Устанавливает идентификационные данные для доступа к X-Server.	**High Level Security** Требуется идентификация **Защита среднего уровня** Требуется идентификация **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** Не требуется	Нет
Права доступа для создания объектов	Задает соответствующее значение для атрибута **umask** в `/etc/security/user`, который указывает права доступа для создания объектов по умолчанию.	**High Level Security** 077 **Защита среднего уровня** 027 **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** 022	Да
Задать размер файла core	Задает соответствующее значение для атрибута **core** в `/etc/security/limits`, который указывает размер файла core для учетной записи root. **Примечание:** Настройки безопасности вступают в силу только для новых сеансов. Настройки безопасности не вступают в силу в тех сеансах, в которых были заданы эти настройки.	**High Level Security** 0 **Защита среднего уровня** 0 **Защита низкого уровня** 0 **Стандартные настройки AIX** 2097151	Да
Активировать функцию SED	Активирует функцию Отключение работы со стеком и запускает команду **sedmgr** для указанных файлов. **Примечание:** Для того чтобы правило вступило в силу, следует перезагрузить систему.	**High Level Security** setidfiles **Защита среднего уровня** Никакого действия **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** Никакого действия	

Таблица 33. Прочие рекоменадции - Эксперт безопасности AIX (продолжение)

Имя кнопки	Описание	Значение, заданное Эксперт безопасности AIX	Отме- на
Проверка сложности корневого пароля	Обеспечивает сложность корневого пароля. Атрибут dictionlist для root имеет значение `/etc/security/aixpert/dictionary/English`, таким образом, команда **passwd** может гарантировать сложность установленного корневого пароля.	**High Level Security** Да **Защита среднего уровня** Да **Защита низкого уровня** Никакого действия **Стандартные настройки AIX** Никакого действия	Да

Отмена настроек безопасности Эксперта безопасности AIX

Некоторые настройки и правила системы безопасности Эксперт безопасности AIX можно отменить.

Следующие правила и параметры защиты Эксперт безопасности AIX нельзя отменить:

- Проверка определений паролей для высокого, среднего и низкого уровня безопасности
- Проверка определений пользователей для высокого, среднего и низкого уровня безопасности
- Проверка определений групп для высокого, среднего и низкого уровня безопасности
- Обновление TCB для высокого, среднего и низкого уровня безопасности
- Включение доступа к X-Server для высокого, среднего и низкого уровня безопасности
- Удаление точки из пути учетных записей, кроме учетной записи root для высокого уровня безопасности и стандартных настроек AIX
- Удаление учетной записи guest для высокого, среднего и низкого уровня безопасности

Проверка системы безопасности Эксперта безопасности AIX

Эксперт безопасности AIX может создавать отчеты о текущих настройках безопасности системы и сети.

После настройки системы с помощью Эксперт безопасности AIX (команда `aixpert`), можно создавать отчеты о различных настройках с помощью опции Проверка системы безопасности. Если если какие-либо из этих настроек были изменены без участия Эксперт безопасности AIX, эти изменения будут отражены в файле `/etc/security/aixpert/check_report.txt` при использовании опции Проверка системы безопасности Эксперт безопасности AIX.

Например, был отключен демон **talkd** в `/etc/inetd.conf`, при применении низкого уровня безопасности. Если позднее демон **talkd** будет включен, то при запуске Проверки системы безопасности, информация об этом будет отражена в файле `check_report.txt` следующим образом:

```
coninetdconf.ksh: Service talk using protocol udp should be disabled, however it is enabled now.
```

Если настройки системы не менялись, файл `check_report.txt` будет пуст.

Опцию Проверка системы безопасности следует периодически запускать, для проверки, изменения настроек, произведенных после того, как система безопасности была настроена с помощью Эксперт безопасности AIX. Опцию Проверка системы безопасности также следует запускать при значительных изменениях системы, таких как установка или обновление программного обеспечения.

Информация, связанная с данной:

команда aixpert

Файлы Эксперта безопасности AIX

Эксперт безопасности AIX создает и использует ряд файлов.

/etc/security/aixpert/core/aixpertall.xml
Содержит список XML всех возможных настроек безопасности.

/etc/security/aixpert/core/appliedaixpert.xml
Содержит список XML применяемых настроек безопасности.

/etc/security/aixpert/core/secaixpert.xml
Содержит список XML выбранных настроек безопасности при обработке Эксперт безопасности AIX GUI.

/etc/security/aixpert/log/aixpert.log
Содержит протокол трассировки применяемых настроек безопасности. Эксперт безопасности AIX не использует протокол syslog; Эксперт безопасности AIX добавляет записи прямо в `/etc/security/aixpert/log/aixpert.log`.

Примечание: Файлы протоколов и Эксперт безопасности AIX XML создаются со следующими правами доступа:

/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-------`

Сценарий для высокого уровня безопасности Эксперта безопасности AIX

Этот сценарий предназначен для Высокого уровня безопасности Эксперт безопасности AIX.

Уровни защиты Эксперт безопасности AIX взяты из документации Национального института стандартов и технологий (NIST) *Программа справочных таблиц настроек безопасности для IT продуктов - руководство для пользователей справочных таблиц и разработчиков* (публикация находится на Web-сайте NIS: http://www.nist.gov/index.html). Однако, разные люди по-разному понимают значение высокого, среднего и низкого уровней безопасности. Важно знать среду, в которой работает ваша система. Если выбрать слишком высокий уровень защиты, можно заблокировать для себя доступ к компьютеру. Если выбрать слишком низкий уровень защиты, ваш компьютер может стать легко уязвимым для сетевых атак.

Здесь приведен пример среды, для которой требуется высокий уровень безопасности. Боб разместил свою систему в аппаратной провайдера служб Internet. Система напрямую подключена к Internet, работает, как сервер HTTP, содержит пользовательские данные и Бобу необходимо иметь возможность управлять ей удаленно. Перед тем как включить систему для работы с ISP, необходимо ее запустить и протестировать в изолированной локальной сети.

Для данной среды подходит высокий уровень защиты, но Бобу необходим удаленный доступ к этой системе. При высоком уровне защиты не доступны **telnet**, **rlogin**, **ftp** и другие обычные соединения, передающие пароли по сети в незашифрованном виде. Такие пароли можно легко отследить через Internet. Бобу нужен безопасный метод для удаленного входа в систему, такой как **openssh**. При изучении документации по Эксперт безопасности AIX, Боб может определить, могут ли какие-либо компоненты оказаться заблокированными, если установить высокий уровень безопасности. Он может отменить выбор этих компонентов в подробная панели высокого уровня защиты. Боб также должен настроить и запустить сервер HTTP и любые другие службы, которые будут работать в его системе.

Когда потом Боб выберет высокий уровень защиты, Эксперт безопасности AIX определит, что запущенные службы необходимы для работы системы и не заблокирует доступ к их портам. Все другие порты могут быть уязвимыми и при высоком уровне безопасности доступ к ним будет заблокирован. После проверки этих настроек, система Боба готова для работы в Internet.

Сценарий для среднего уровня безопасности Эксперта безопасности AIX

Этот сценарий предназначен для Среднего уровня безопасности Эксперт безопасности AIX.

Элис необходимо обеспечить безопасность системы, которая будет подключена к корпоративной сети, в которой установлен брандмауэр. Сеть надежно защищена. Этой системой будет пользоваться большое количество пользователей, которым необходимо иметь доступ к системе через **telnet** и **ftp**. Элис хочет установить общие настройки безопасности, такие как защита от сканирования портов и пароли, с истекающими сроками действия, но при этом доступ к системе должен быть открыт для большинства методов удаленного доступа. В этом случае, для системы Элис лучше всего подходит средний уровень безопасности.

Сценарий для низкого уровня безопасности Эксперта безопасности AIX

Этот сценарий предназначен для Низкого уровня безопасности Эксперт безопасности AIX.

Брюс управляет системой в течение некоторого времени. Система располагается в изолированной защищенной локальной сети. Эту систему используют большое количество пользователей и служб. Он хочет установить наименьший уровень защиты, не прерывая доступ к системе. Для системы Брюса лучше всего подходит низкий уровень безопасности.

Справочная таблица по защите

В этом разделе содержится справочная таблица со списком действий для обеспечения защиты новой или уже существующей системы.

Хотя этот список и не полон, он может использоваться в качестве основы для создания справочной таблицы по защите конкретной системы.

- Устанавливайте AIX с надежного базового носителя. Во время установки выполните следующие действия:
  - Не устанавливайте на серверах приложения рабочего стола, такие как CDE, GNOME или KDE.
  - Установите все рекомендуемые исправления, связанные с защитой. Последние бюллетени служб, советы по безопасности и информацию об исправлениях можно найти на веб-сайте IBM System p eServer Support Fixes (http://www.ibm.com/support/fixcentral).
  - Создайте резервную копию системы сразу после установки и сохраните эту копию в надежном месте.
- Создайте списки управления доступом к важным файлам и каталогам.
- Отключите ненужные пользовательские и системные учетные записи, например, daemon, bin, sys, adm, lp и uucp. Удалять эти записи не рекомендуется, поскольку хранящаяся в них информация (например, ИД и имена пользователей) может понадобиться при восстановлении данных из резервной копии. Если после

создания нового пользователя с тем же именем будет восстановлена резервная копия, то новый пользователь может получить нежелательный доступ к системе.

- Регулярно просматривайте файлы `/etc/inetd.conf`, `/etc/inittab`, `/etc/rc.nfs` и `/etc/rc.tcpip` и удаляйте все ненужные демоны и службы.
- Убедитесь, что права доступа к следующим файлам заданы правильно:

```
-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
```

- Запретите удаленный вход пользователя root. Этот пользователь должен иметь возможность входить в систему только с консоли.
- Включите контроль системы. Дополнительная информация приведена в разделе “Обзор подсистемы контроля” на стр. 136.
- Включите стратегию управления входом в систему. Дополнительная информация приведена в разделе “Управление входом в систему” на стр. 34.
- Запретите пользователям вызывать команду `xhost`. Дополнительная информация приведена в разделе “Рекомендации по работе с X11 и CDE” на стр. 40.
- Запретите изменять переменную среды **PATH**. Дополнительная информация приведена в разделе “Переменная среды PATH” на стр. 56.
- Отключите telnet, rlogin и rsh. Дополнительная информация приведена в разделе “Защита TCP/IP” на стр. 209.
- Включите управление учетными записями пользователей. Дополнительная информация приведена в разделе “Управление учетными записями пользователей” на стр. 53.
- Реализуйте жесткую стратегию создания паролей. Дополнительная информация приведена в разделе “Пароль” на стр. 64.
- Установите дисковые квоты для пользователей. Дополнительная информация приведена в разделе “Восстановление после превышения квоты” на стр. 77.
- Разрешите вызывать команду **su** только администраторам. Отслеживайте протокол применения команды **su** в файле `/var/adm/sulog`.
- Включите блокировку экрана при работе с X-Windows.
- Разрешите доступ к командам **cron** и **at** только тем пользователям, которым это действительно необходимо.
- Создайте псевдоним команды **ls**, показывающий скрытые файлы и символы в именах файлов.
- Создайте псевдоним команды **rm**, позволяющий избежать случайного удаления системных файлов.
- Отключите ненужные сетевые службы. Дополнительная информация приведена в разделе “Сетевые службы” на стр. 217.
- Регулярно создавайте резервные копии системы и проверяйте их целостность.
- Подпишитесь на списки рассылки, связанные с защитой системы.

Обзор основных системных служб AIX

В следующей таблице перечислены основные системные службы AIX. Эта таблица поможет вам определить, с чего следует начать при настройке средств защиты системы.

Перед настройкой защиты системы сохраните все исходные файлы конфигурации, в первую очередь следующие:

- `/etc/inetd.conf`
- `/etc/inittab`
- `/etc/rc.nfs`

- `/etc/rc.tcpip`

Служба	Демон	Кем запускается	Функция	Комментарий
inetd/bootps	inetd	/etc/inetd.conf	Служба загрузки бездисковых клиентов.	• Необходима для NIM и удаленной загрузки систем. • Работает вместе с tftp. • В большинстве случаев следует отключить.
inetd/chargen	inetd	/etc/inetd.conf	Генератор символов (только для тестирования).	• Может применяться в качестве службы TCP и UDP. • Открывает возможность атак типа "отказ в обслуживании". • Отключите, если вы не тестируете сеть.
inetd/cmsd	inetd	/etc/inetd.conf	Служба календаря (используется CDE).	• Работает под управлением ИД root, что представляет потенциальную угрозу защите. • Если эта служба не нужна CDE, то отключите ее. • Отключите на серверах баз данных.
inetd/comsat	inetd	/etc/inetd.conf	Уведомляет о получении электронной почты.	• Работает под управлением ИД root, что представляет потенциальную угрозу защите. • Требуется редко. • Отключите.
inetd/daytime	inetd	/etc/inetd.conf	Устаревшая служба времени (только для тестирования).	• Работает под управлением ИД root. • Может применяться в качестве службы TCP и UDP. • Открывает возможность атак типа "отказ в обслуживании" с помощью PING. • Служба устарела и применяется только для тестирования. • Отключите.
inetd/discard	inetd	/etc/inetd.conf	Служба /dev/null (только для тестирования).	• Может применяться в качестве службы TCP и UDP. • Позволяет осуществлять атаки типа "отказ в обслуживании". • Служба устарела и применяется только для тестирования. • Отключите.
inetd/dtspc	inetd	/etc/inetd.conf	Управление подпроцессом CDE.	• Эта служба автоматически запускается демоном **inetd** в ответ на запрос клиента CDE на запуск процесса на хосте демона. При этом возникает потенциальная угроза атаки. • Отключите на серверах без CDE. • CDE может работать без этой службы. • Отключите, если нет крайней необходимости в этой службе

Служба	Демон	Кем запускается	Функция	Комментарий
inetd/echo	inetd	etc/inetd.conf	Служба эхо (только для тестирования).	• Может применяться в качестве службы TCP и UDP. • Позволяет осуществлять атаки типа "отказ в обслуживании" и "Smurf". • Применяется для передачи эхо-пакетов на чей-либо адрес для проникновения через брандмауэр или атаки путем передачи чрезмерного объема данных. • Отключите.
inetd/exec	inetd	/etc/inetd.conf	Служба удаленного выполнения.	• Работает под управлением ИД root. • Требует ввода ИД пользователя и пароля, которые передаются без шифрования. • Эта служба чрезвычайно неустойчива к перехвату. • Отключите.
inetd/finger	inetd	/etc/inetd.conf	Предоставляет информацию о пользователях.	• Работает под управлением ИД root. • Предоставляет информацию о системах и пользователях. • Отключите.
inetd/ftp	inetd	/etc/inetd.conf	Протокол передачи файлов.	• Работает под управлением ИД root. • ИД пользователей и пароли передаются без шифрования, что позволяет перехватывать их. • Отключите эту службу и применяйте общедоступные защищенные пакеты.
inetd/imap2	inetd	/etc/inetd.conf	Почтовый протокол Internet.	• Применяйте только последнюю версию этого сервера. • Необходима только при использовании системы в качестве почтового сервера. В противном случае необходимо отключить эту службу. • ИД пользователей и пароли передаются без шифрования.
inetd/klogin	inetd	/etc/inetd.conf	Вход в систему Kerberos.	• Включена, если в вашей системе используется идентификация Kerberos.
inetd/kshell	inetd	/etc/inetd.conf	Оболочка Kerberos.	• Включена, если в вашей системе используется идентификация Kerberos.
inetd/login	inetd	/etc/inetd.conf	Служба удаленного входа в систему rlogin.	• Неустойчива к перехвату IP и DNS. • Данные, включая ИД и пароли пользователей, передаются без шифрования. • Работает под управлением ИД root. • Вместо этой службы следует применять защищенную оболочку.
inetd/netstat	inetd	/etc/inetd.conf	Просмотр сведений о текущем состоянии сети.	• Может предоставить хакерам информацию о сети. • Отключите.

Служба	Демон	Кем запускается	Функция	Комментарий
inetd/ntalk	inetd	/etc/inetd.conf	Позволяет пользователям обмениваться сообщениями.	• Работает под управлением ИД root. • Не требуется на рабочих или базовых серверах. • Отключите, если нет крайней необходимости в этой службе
inetd/pcnfsd	inetd	/etc/inetd.conf	Служба поддержки сетевой файловой системы PC.	• Если эта служба не используется, отключите ее. • Если такая служба необходима, то рекомендуется применять службу Samba, поскольку демон pcnfsd является предшественником разработанной фирмой Microsoft спецификации SMB.
inetd/pop3	inetd	/etc/linetd.conf	Протокол доступа к почтовому серверу.	• ИД и пароли пользователей передаются без шифрования. • Необходима только в том случае, если система применяется в качестве почтового сервера и обслуживает клиентов, поддерживающих только протокол POP3 • Вместо POP3 рекомендуется применять IMAP, если клиенты поддерживают IMAP, либо POP3s. Данная служба поддерживает туннели SSL. • Отключите эту службу, если система не применяется в качестве почтового сервера или не обслуживает клиентов, поддерживающих только POP.
inetd/rexd	inetd	/etc/inetd.conf	Удаленное выполнение команд.	• Работает под управлением ИД root. • Взаимодействует с командой **on**. • Отключите. • Вместо этой службы рекомендуется применять **rsh** и **rshd**.
inetd/quotad	inetd	/etc/inetd.conf	Возвращает информацию о файловых квотах для клиентов NFS.	• Необходима только при использовании служб NFS. • Отключите эту службу, если она не требуется для отправки ответов на команду **quota** • Если эта служба необходима, то следует регулярно устанавливать выпускаемые для нее исправления.
inetd/rstatd	inetd	/etc/inetd.conf	Служба сбора статистики ядра.	• Для отслеживания систем рекомендуется применять SNMP, а эту службу отключить. • Данная служба необходима для работы команды **rup**.
inetd/rusersd	inetd	/etc/inetd.conf	Передает информацию о работающих в системе пользователях.	• Это второстепенная служба. Отключите. • Работает под управлением ИД root. • Предоставляет команде rusers список работающих в системе пользователей.

Служба	Демон	Кем запускается	Функция	Комментарий
inetd/rwalld	inetd	/etc/inetd.conf	Передает сообщение всем пользова- телям.	• Работает под управлением ИД root. • Если система используется пользователями в интерактивном режиме, то эта служба может вам потребоваться. • В рабочих системах и на серверах баз данных эта служба не нужна. • Отключите.
inetd/shell	inetd	/etc/inetd.conf	Служба удаленной оболочки rsh.	• Отключите эту службу. Вместо данной службы рекомендуется применять защищенную оболочку. • Если необходимо применять именно эту службу, то используйте обрамляющие структуры TCP, чтобы избежать перехвата данных и снизить опасность нарушения защиты. • Необходима работы **Xhier**.
inetd/sprayd	inetd	/etc/inetd.conf	Проверка связи с помощью RPC и команды spray.	• Работает под управлением ИД root. • Может потребоваться для диагностики неполадок NFS. • Отключите эту службу, если вы не применяете NFS.
inetd/systat	inetd	/etc/inted.conf	Отчет "ps -ef".	• Позволяет удаленным системам просматривать информацию о состоянии процессов в вашей системе. • По умолчанию эта служба отключена. Необходимо проверять ее периодически и убеждаться, что она не включена.
inetd/talk	inetd	/etc/inetd.conf	Устанавливает сеанс связи между двумя пользовате- лями в сети.	• Это не обязательная служба. • Применяется вместе с командой **talk**. • Применяет порт UDP 517. • Отключите эту службу, если вам не требуется устанавливать сеансы диалога между интерактивными пользователями UNIX
inetd/ntalk	inetd	/etc/inetd.conf	Новый вариант команды talk. Устанавливает сеанс связи между двумя пользовате- лями в сети.	• Это не обязательная служба. • Применяется вместе с командой **talk**. • Применяет порт UDP 517. • Отключите эту службу, если вам не требуется устанавливать сеансы диалога между интерактивными пользователями UNIX
inetd/telnet	inetd	/etc/inetd.conf	Служба telnet.	• Поддерживает удаленный вход в систему. ИД и пароли пользователей передаются без шифрования. • Если это возможно, отключите данную службу и используйте вместо нее защищенную оболочку.

Служба	Демон	Кем запускается	Функция	Комментарий
inetd/tftp	inetd	/etc/inetd.conf	Упрощенный протокол передачи файлов.	• Применяет порт UDP 69. • Работает под управлением ИД root. • Применяется NIM. • Отключите эту службу, если вы не применяете NIM и не используете систему для загрузки бездисковых рабочих станций.
inetd/time	inetd	/etc/inetd.conf	Устаревшая служба времени.	• Внутренняя функция демона **inetd**, применяемая командой **rdate**. • Может применяться в качестве службы TCP и UDP. • Иногда применяется для синхронизации системных часов при загрузке. • Устаревшая служба. Вместо нее рекомендуется применять ntpdate • Отключите эту службу и протестируйте работу своих систем (загрузка/перезагрузка). Если неполадок не обнаружено, не включайте эту службу.
inetd/ttdbserver	inetd	/etc/inetd.conf	Сервер баз данных tool-talk (для CDE).	• **rpc.ttdbserverd** работает под управлением ИД root. • Запускается как обязательная служба CDE, но CDE может работать и без этой службы. • Не следует применять эту службу на базовых серверах, а также в системах с высокими требованиями к защите.
inetd/uucp	inetd	/etc/inetd.conf	Поддержка сети UUCP.	• Отключите эту службу, если вы не применяете приложения, работающие с UUCP.
inittab/dt	init	Сценарий /etc/rc.dt в /etc/inittab	Вход в систему рабочего стола среды CDE.	• Запускает на консоли сервер X11. • Поддерживает протокол управления дисплеями X11 (xdcmp), обеспечивая возможность входа в ту же систему с других станций X11. • Эта служба должна применяться только на персональных рабочих станциях. Не следует применять ее на базовых серверах.
inittab/dt_nogb	init	/etc/inittab	Вход в систему рабочего стола среды CDE (загрузка без графики).	• Графический дисплей инициализируется лишь после полной загрузки системы. • Ограничения те же, что и у inittab/dt
inittab/httpdlite	init	/etc/inittab	Web-сервер для команды **docsearch**.	• Web-сервер по умолчанию для службы поиска документации. • Отключите, если ваша система не используется в качестве сервера документации.

Служба	Демон	Кем запускается	Функция	Комментарий
inittab/i4ls	init	/etc/inittab	Серверы администра- тора лицензий.	• Включите в системах, используемых для разработки приложений. • Отключите в рабочих системах. • Включите на серверах баз данных, к которым предъявляются особые требования по лицензированию. • Обеспечивает поддержку компиляторов, баз данных и других лицензионных продуктов.
inittab/imqss	init	/etc/inittab	Механизм поиска для службы поиска документации.	• Компонент Web-сервера для службы поиска документации. • Отключите, если ваша система не используется в качестве сервера документации.
inittab/lpd	init	/etc/inittab	Интерфейс построчного принтера BSD.	• Принимает задания печати, поступающие из других систем. • Вы можете отключить эту службу, сохранив возможность отправки заданий на сервер печати. • Отключите эту службу, убедившись, что это не повлияло на функции печати.
inittab/nfs	init	/etc/inittab	Службы сетевой файловой системы и информации о сети.	• Службы NFS и NIS на основе UDP/RPC. • Минимальный объем идентификации. • Отключите службу на базовых серверах.
inittab/piobe	init	/etc/inittab	Базовая программа печати.	• Обрабатывает планирование, буферизацию и печать заданий, переданных на выполнение демоном **qdaemon** • Отключите эту службу, если вы не печатаете документы в своей системе, а отправляете их на сервер печати.
inittab/qdaemon	init	/etc/inittab	Демон очередей (для печати).	• Передает задания печати демону **piobe**. • Отключите эту службу, если вы не печатаете документы в своей системе.
inittab/uprintfd	init	/etc/inittab	Сообщения ядра.	• Обычно эта служба не требуется. • Отключите.
inittab/writesrv	init	/etc/inittab	Отправка сообщений на терминалы.	• Применяется только пользователями интерактивных рабочих станций UNIX • Отключите эту службу на серверах, базовых серверах баз данных и в системах, применяемых для разработки приложений. • Включите эту службу на рабочих станциях.

Служба	Демон	Кем запускается	Функция	Комментарий
inittab/xdm	init	/etc/inittab	Традиционный администратор дисплеев X11.	• Не запускайте на базовых рабочих серверах и серверах баз данных. • Не запускайте в системах для разработки приложений, если не требуются функции управления дисплеями X11. • Может применяться на рабочих станциях, если необходима поддержка графического режима.
rc.nfs/automountd		/etc/rc.nfs	Авт. монтирование файловых систем.	• Включите эту службу на рабочих станциях, если вы применяете NFS. • Не включайте службу автоматического монтирования на базовых серверах и в системах, применяемых для разработки приложений.
rc.nfs/biod		/etc/rc.nfs	Демон блокового ввода-вывода (необходим для сервера NFS).	• Включите для поддержки сервера NFS. • Если вы не работаете с сервером NFS, то отключите эту службу, а также nfsd и rpc.mountd
rc.nfs/keyserv		/etc/rc.nfs	Защищенный сервер ключей RPC.	• Управляет ключами, необходимыми для обеспечения защиты RPC. • Отключите эту службу, если вы *не* используете NFS и NIS.
rc.nfs/nfsd		/etc/rc.nfs	Службы NFS (необходимы для сервера NFS).	• Слабые средства идентификации. • Может привести к краху стека. • Включите на файловых серверах, использующих NFS. • Если вы отключили эту службу, то отключите также **biod**, **nfsd** и **rpc.mountd**.
rc.nfs/rpc.lockd		/etc/rc.nfs	Блокировка файлов NFS.	• Отключите, если вы не применяете NFS. • Отключите эту службу, если вы не применяете блокировку файлов в сети. • Демон **lockd** представляет потенциальную угрозу защите системы, он упомянут в списке SANS Top Ten Security Threats.
rc.nfs/rpc.mountd		/etc/rc.nfs	Монтирование файлов NFS (служба необходима для сервера NFS).	• Слабые средства идентификации. • Может привести к краху стека. • Следует включить на файловых серверах, применяющих NFS. • Если вы отключили эту службу, то отключите также **biod** и **nfsd**.
rc.nfs/rpc.statd		/etc/rc.nfs	Блокировка файлов NFS (для восст.).	• Реализует блокировку файлов в NFS. • Отключите, если вы не применяете NFS.

Служба	Демон	Кем запускается	Функция	Комментарий
rc.nfs/rpc.yppasswdd		/etc/rc.nfs	Демон паролей NIS (для сервера NIS).	• Применяется для управления локальным файлом паролей. • Эта служба необходима только на сервере NIS. Отключите ее на всех остальных системах.
rc.nfs/ypupdated		/etc/rc.nfs	Демон обновления NIS (для подчиненных серверов NIS).	• Получает образы баз данных NIS с сервера NIS. • Эта служба необходима только на подчиненном сервере главного сервера NIS.
rc.tcpip/autoconf6		/etc/rc.tcpip	Интерфейсы IPv6.	• Отключите, если вы не работаете с IPv6
rc.tcpip/dhcpcd		/etc/rc.tcpip	Протокол динамической настройки хостов (клиент).	• Базовые серверы не должны выполнять функции промежуточных агентов DHCP. Отключите эту службу. • Если хост не применяет DHCP, отключите эту службу.
rc.tcpip/dhcprd		/etc/rc.tcpip	Протокол динамической настройки хостов (промежу- точный агент).	• Получает широковещательные запросы DHCP и отправляет их на сервер в другой сети. • Дублирует функции аналогичной службы маршрутизаторов. • Отключите эту службу, если вы не применяете DHCP или не передаете информацию между сетями.
rc.tcpip/dhcpsd		/etc/rc.tcpip	Протокол динамической настройки хостов (сервер).	• Отвечает на запросы DHCP, отправляемые клиентами при загрузке; передает клиентам такую информацию, как имя, IP-адрес, номер, маска сети, а также адрес маршрутизатора и широковещательный адрес. • Отключите эту службу, если вы не применяете DHCP. • Отключите на рабочих и базовых серверах, а также на хостах, не использующих DHCP.
rc.tcpip/dpid2		/etc/rc.tcpip	Устаревшая служба SNMP.	• Отключите, если вам не требуется поддержка SNMP.
rc.tcpip/gated		/etc.rc.tcpip	Маршрутиза- ция между интерфейсами.	• Обеспечивает функции маршрутизации. • Отключите эту службу и применяйте вместо нее RIP или маршрутизатор.
rc.tcpip/inetd		/etc/rc.tcpip	Службы inetd.	• Для обеспечения максимальной защиты системы эти службы необходимо отключить, однако в большинстве случаев это невозможно. • Отключение приведен к запрету служб удаленных оболочек, необходимых для некоторых почтовых и Web-серверов.

Служба	Демон	Кем запускается	Функция	Комментарий
rc.tcpip/mrouted		/etc/rc.tcpip	Маршрутиза- ция многоцелевой рассылки.	• Эмулирует функции маршрутизатора по рассылке пакетов многоцелевой рассылки между сегментами сети. • Отключите эту службу. Вместо нее воспользуйтесь маршрутизатором.
rc.tcpip/names		/etc/rc.tcpip	Сервер имен (DNS).	• Включите эту службу только в том случае, если система применяется в качестве сервера имен DNS. • Отключите эту службу на рабочих станциях, системах, применяемых для разработки приложений, а также на рабочих серверах.
rc.tcpip/ndp-host		/etc/rc.tcpip	Служба хоста IPv6.	• Отключите, если вы не применяете IPv6
rc.tcpip/ndp-router		/etc/rc.tcpip	Маршрутиза- ция IPv6.	• Отключите, если вы не работаете с IPv6. Вместо IPv6 воспользуйтесь маршрутизатором.
rc.tcpip/portmap		/etc/rc.tcpip	Сервер списка служб RPC.	• Необходимая служба. • Серверы RPC регистрируются демоном **portmap**. Клиенты, которым необходимо обратиться к службе RPC, обращаются к демону **portmap**, который им сообщает, как обратиться к той или иной службе. • Отключите службу **portmap** только в том случае, если вы отказались от всех остальных служб RPC.
rc.tcpip/routed		/etc/rc.tcpip	Маршрутиза- ция RIP между интерфейсами.	• Обеспечивает функции маршрутизации. • Отключите, если у вас есть маршрутизатор для пересылки пакетов между сетями.
rc.tcpip/rwhod		/etc/rc.tcpip	Удаленный демон команды "who".	• Собирает данные и рассылает их серверам той же сети. • Отключите эту службу.
rc.tcpip/sendmail		/etc/rc.tcpip	Почтовая служба.	• Работает под управлением ИД root. • Отключите эту службу, если система не применяется в качестве почтового сервера. • Если служба отключена, то выполните одно из следующих действий: – Добавьте запись в crontab для очистки очереди. Укажите команду **/usr/lib/sendmail -q**. – Настройте службы DNS таким образом, чтобы почта вашей системы доставлялась в какую-либо другую систему.
rc.tcpip/snmpd		/etc/rc.tcpip	Простой протокол управления сетью.	• Отключите эту службу, если вы не отслеживаете состояние системы с помощью инструментов SNMP. • Наличие SNMP может быть существенным для важных серверов.

Служба	Демон	Кем запускается	Функция	Комментарий
rc.tcpip/syslogd		/etc/rc.tcpip	Системный протокол событий.	• Выключать эту службу *не* рекомендуется. • Подвержена атакам типа "отказ в обслуживании". • Необходима во всех системах.
rc.tcpip/timed		/etc/rc.tcpip	Устаревший демон времени.	• Отключите эту службу и применяйте вместо нее xntp.
rc.tcpip/xntpd		/etc/rc.tcpip	Новый демон времени.	• Обеспечивает синхронизацию часов в системах. • Отключите эту службу. • Настройте некоторые системы в качестве серверов времени и разрешите остальным системам синхронизировать время по серверам с помощью заданий cron, вызывающих ntpdate.
dt login		/usr/dt/config/Xaccess	CDE без ограничений.	• Если вы не обеспечиваете вход в систему CDE группе станций X11, то вы можете ограничить dtlogin для консоли.
Анонимная служба FTP		user rmuser -p <имя>	Анонимный ftp.	• Анонимный FTP не обеспечивает контроля за операциями отдельных пользователей, выполняемых с помощью FTP. • Если существует учетная запись пользователя ftp, удалите ее с помощью команды **rmuser -p ftp**. • Для повышения надежности защиты можно перечислить в файле `/etc/ftpusers` пользователей, которым должен быть запрещен доступ к системе с помощью FTP.
Запись с помощью анонимного FTP.			Передача файлов на сервер с помощью анонимного FTP.	• Пользователю ftp не должны принадлежать какие-либо файлы. • Загрузка файлов с помощью анонимного FTP потенциально разрешает размещение в системе программ, созданных злоумышленниками. • Перечислите в файле `/etc/ftpusers` пользователей, которым необходимо запретить доступ. • Примеры системных пользователей, которым следует запретить загрузку файлов в систему с помощью анонимного FTP: root, daemon, bin.sys, admin.uucp, guest, nobody, lpd, nuucp, ladp. • Измените владельца и группу для файла `ftpusers` с помощью следующей команды: `chown root:system /etc/ftpusers` • Измените права доступа к файлу `ftpusers` с помощью следующей команды: `chmod 644 /etc/ftpusers`

Служба	Демон	Кем запускается	Функция	Комментарий
ftp.restrict			Доступ к системным учетным записям с помощью FTP.	• С помощью файла `ftpusers` следует запретить всем внешним пользователям заменять файлы в корневом каталоге
root.access		/etc/security/user	Подключение к учетной записи root с помощью rlogin/telnet.	• Присвойте опции rlogin в файле `etc/security/user` значение false. • Любой пользователь, входящий в систему под именем root, должен сначала войти под собственным именем и лишь затем переключиться на учетную запись root с помощью команды **su**; это обеспечит необходимый контрольный след.
snmpd.readWrite		/etc/snmpd.conf	Поддержка SNMP readWrite.	• Если вы *не* применяете SNMP, то отключите демон SNMP. • Запретите связи систем и частные связи с помощью файла `/etc/snmpd.conf`. • Для связи 'public' разрешите только те IP-адреса, с которых отслеживается ваша система.
syslog.conf			Настройка syslogd.	• Если вы не настроили `/etc/syslog.conf`, то запретите запуск этого демона. • Если вы применяете `syslog.conf` для ведения протокола системных сообщений, то включите эту службу.

Обзор сетевых опций

Некоторые сетевые опции непосредственным образом влияют на степень защищенности сети от возможных атак. Каждая такая опция отключает (`0`) или включает (`1`) определенный режим обработки сетевых пакетов. В следующем разделе перечислены опции, которые можно использовать с командой **no**.

Параметр	Команда	Назначение
bcastping	/usr/sbin/no -o bcastping=0	Разрешает отвечать на эхозапросы ICMP, передаваемые в широковещательном режиме. Отключив этот режим, можно защититься от возможных атак типа Smurf (атака, направленная на отказ в обслуживании путем отправки большого количества широковещательных запросов на конкретный IP-адрес).
clean_partial_conns	/usr/sbin/no -o clean_partial_conns=1	Указывает, включена ли защита от возможных атак типа SYN (атаки, направленные на отказ в обслуживании путем отправки большого числа пакетов SYN и вынуждение браузера выделять память для установки потенциальных сеансов связи с клиентами).
directed_broadcast	/usr/sbin/no -o directed_broadcast=0	Указывает, разрешена ли прямая отправка широковещательных пакетов через шлюз. Если этому значению будет присвоено значение 0, такие пакеты не будут передаваться в удаленную сеть.

Параметр	Команда	Назначение
icmpaddressmask	/usr/sbin/no -o icmpaddressmask=0	Указывает, будет ли система отвечать на запросы маски подсети по протоколу ICMP. Отключив этот режим, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.
ipforwarding	/usr/sbin/no -o ipforwarding=0	Указывает, будет ли ядро выполнять пересылку пакетов. Отключив этот режим, можно запретить перенаправление пакетов в другие сети.
ipignoreredirects	/usr/sbin/no -o ipignoreredirects=1	Указывает способ обработки запросов на перенаправление пакетов.
ipsendredirects	/usr/sbin/no -o ipsendredirects=0	Указывает, будет ли ядро передавать сигналы о перенаправлении пакетов. Отключив этот режим, можно запретить перенаправление пакетов в другие сети.
ip6srcrouteforward	/usr/sbin/no -o ip6srcrouteforward=0	Указывает, будет ли система пересылать пакеты IPv6, отправляемые по сложным маршрутам ICMP. Отключив этот режим, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.
ipsrcrouteforward	/usr/sbin/no -o ipsrcrouteforward=0	Указывает, будет ли система пересылать пакеты, отправляемые по сложным маршрутам ICMP. Отключив этот режим, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.
ipsrcrouterecv	/usr/sbin/no -o ipsrcrouterecv=0	Указывает, будет ли система принимать пакеты, отправляемые по сложным маршрутам. Отключив этот режим, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.
ipsrcroutesend	/usr/sbin/no -o ipsrcroutesend=0	Указывает, разрешено ли приложениям отправлять пакеты по сложным маршрутам ICMP. Отключив этот режим, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.
nonlocsroute	/usr/sbin/no -o nonlocsrcroute=0	Указывает, разрешена ли отправка пакетов IP по сложным маршрутам за пределы локальной сети. Отключив этот режим, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.
tcp_icmpsecure	/usr/sbin/no -o tcp_icmpsecurer=1	Защищает соединения TCP от атак по ICMP (протокол управления Internet-сообщениями) с подавлением источника и атак PMTUD (вычисление MTU маршрута). Проверяет полезную нагрузку ICMP-сообщения, определяя, находится ли порядковый номер заголовка TCP в диапазоне допустимых порядковых номеров. Допустимые значения: 0=off (значение по умолчанию); 1=on.
ip_nfrag	/usr/sbin/no -o ip_nfrag=200	Указывает максимальное количество фрагментов пакета IP, которые могут храниться в очереди сборки IP одновременно (значение по умолчанию равно 200 - в очереди сборки IP одновременно могут находиться до 200 фрагментов IP-пакета).

Параметр	Команда	Назначение
tcp_pmtu_discover	/usr/sbin/no -o tcp_pmtu_discover=0	Отключив этот режим, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.
tcp_tcpsecure	/usr/sbin/no -o tcp_tcpsecure=7	Защищает уязвимые места соединения TCP. Допустимые значения: 0=нет защиты; 1=отправка установленному соединению фиктивного SYN; 2=отправка установленному соединению фиктивного RST; 3=ввод данных в установленное соединение TCP; 5–7=комбинация вышеперечисленных уязвимых мест.
udp_pmtu_discover	/usr/sbin/no -o udp_pmtu_discover=0	Включает или выключает режим определения MTU для приложений протокола TCP. Отключив этот режим, можно защититься от возможных атак, использующих знание о конкретных маршрутах в вашей сети.

Дополнительная информация о настраиваемых сетевых опциях приведена в разделе *Руководство по настройке производительности*.

Trusted AIX

Trusted AIX обеспечивает функции многоуровневой защиты (MLS) в AIX.

Примечание: MLS также называется защитой на основе меток.

В отличие от стандартных средств защиты в AIX, в Trusted AIX защита реализована на основе меток, присваиваемым всем субъектам и объектам в системе.

Примечание: При установке Trusted AIX включается среда защиты на основе меток AIX. Управление доступом в системе выполняется на основе меток, реализующих среду многоуровневой защиты (MLS) с поддержкой следующих сущностей:

- Объекты с метками: файлы, объекты IPC, сетевые пакеты и прочие объекты с метками
- Принтеры с метками
- Защищенная сеть: поддержка RIPSO и CIPSO в IPv4 и IPv6

Обратите внимание, что при выборе этого режима установки для восстановления обычной среды AIX потребуется выполнить установку AIX с заменой всех данных. Определите, требуется ли вам среда Trusted AIX, перед выбором этого режима установки. Дополнительная информация о Trusted AIX приведена в общедоступной документации AIX.

Функции защиты стандартного AIX обеспечивают основные потребности защиты системы и сети. Основные функции защиты AIX включают следующее:

- управление доступом к системе и сети с помощью имени и пароля
- права доступа к файлам для пользователя, группы или для всех
- списки управления доступом
- подсистема контроля
- ролевое управление доступом (RBAC)

Trusted AIX, опираясь на эти основные функции защиты AIX, расширяет возможности защиты AIX для сетевых подсистем.

Trusted AIX совместим с API AIX. Любое приложение, работающее в AIX, также будет работать в Trusted AIX. Однако существуют дополнительные ограничения защиты, и приложения, созданные без учета MLS, должны будут иметь соответствующие права доступа для работы в среде Trusted AIX. Для профилирования приложений в таких сценариях можно использовать команду **tracepriv**.

Trusted AIX расширяет API AIX дополнительными функциями защиты. При этом разработчики могут создавать защищенные приложения с помощью API AIX и расширений Trusted AIX.

Trusted AIX позволяет обрабатывать информацию в системах AIX на нескольких уровнях защиты. Он спроектирован в соответствии с требованиями TCSEC Министерства обороны США и европейскими требованиями ITSEC для расширенной защиты уровня B1.

В книгах Защита основной операционной системы и Защита сети описаны стандартные возможности защиты AIX.

Введение в Trusted AIX

Trusted AIX улучшает защиту стандартной операционной системы AIX за счет поддержки функции защиты на основе меток.

Среда защиты на основе меток может быть установлена во время установки Trusted AIX. Для восстановления обычной среды AIX после установки Trusted AIX потребуется выполнить установку с заменой всех данных. После установки среда Trusted AIX будет применяться для всей системы AIX, включая все WPAR, созданные в AIX. Защита, основанная на метках (также называемая многоуровневой защитой, MLS), широко применяется в военных и разведывательных организациях, но она также может применяться и в коммерческих структурах. Это достигается благодаря настройке меток, предусмотренных в Trusted AIX. При первой установке Trusted AIX создаются метки, соответствующие стандартной реализации MLS.

Среда Trusted AIX создается за счет включения в AIX некоторых дополнительных пакетов и наборов файлов. Кроме того, ядро переводится в режим работы Trusted AIX. При загрузке с CD или DVD система загружается в стандартной среде AIX. В меню установки можно выбрать опцию Trusted AIX, чтобы установить файлы для MLS. По окончании установки программа установки выполняет первую перезагрузку системы. В ходе первой перезагрузки Помощник по настройке открывает меню для настройки пользователей, при этом настраиваются пользователи ISSO, SA и SO, и после завершения загрузки создается среда MLS.

Улучшенная защита системы с помощью Trusted AIX обеспечивается благодаря следующим главным принципам защиты информации:

- Конфиденциальность
- Целостность
- Готовность
- Отчетность

В добавление к функциям защиты AIX в Trusted AIX предусмотрены следующие функции:

Метки секретности (SL)
: Всем процессам и файлам присваиваются метки согласно их уровню защиты. Процессы могут иметь доступ только к объектам в своей области защиты.

Метки целостности (TL)
: Всем процессам и файлам присваиваются метки согласно их уровню целостности. Файлы не могут записываться процессами, уровень меток целостности которых задан ниже, чем файл. Процессы не могут читать из файлов, уровень меток целостности которых задан ниже, чем процесс.

Флаги защиты файлов
: Отдельные файлы могут иметь дополнительные флаги, управляющие операциями, связанными с защитой.

Флаги защиты ядра
Некоторые функции защиты могут включаться или выключаться для всей системы.

Привилегии
Многие команды и системные вызовы доступны только процессам с определенными привилегиями.

Права доступа
Каждому пользователю могут предоставляться индивидуальные права доступа. Права доступа разрешают пользователю выполнять определенные операции, связанные с защитой. Права доступа предоставляются на основе ролей.

Роли Функция ролевого управления доступом является частью Trusted AIX и обеспечивает выборочное предоставление административных прав пользователям, не имеющим прав доступа администратора. Это достигается за счет объединения набора прав доступа в роли и связывания пользователей и ролей.

Конфиденциальность

Это понятие относится к предотвращению угроз, связанных с несанкционированным доступом к секретной информации.

Trusted AIX предоставляет механизмы повторного использования объектов и управления доступом для защиты всех ресурсов данных. Операционная система следит за тем, чтобы доступ к защищенным данным имели только пользователи с соответствующими правами доступа, и чтобы эти пользователи не могли предоставить доступ к таким ресурсам прочим пользователям, случайно или намеренно.

Администраторы могут запретить копирование важных файлов на дискеты или прочие съемные носители, или по сети. Эта мера защиты обеспечивается операционной системой, и ее не могут обойти злоумышленники или вредоносные процессы.

Целостность

Это понятие относится к предотвращению угроз, связанных с несанкционированным изменением информации.

В Trusted AIX предусмотрены различные механизмы, обеспечивающие целостность защищенной компьютерной базы и данных, созданных внутри системы или импортированных по сети. Различные механизмы контроля прав доступа защищают информацию от несанкционированного изменения. Для того чтобы злоумышленники или вредоносные процессы не могли получить доступ к системным ресурсам или выключить их, в Trusted AIX выключены права доступа root. Вместо работы с правами root реализовано разделение обязанностей администратора на несколько административных прав доступа и ролей.

Готовность

Это понятие относится к предотвращению угроз, связанных с нарушением работы служб системы. Например, если вредоносная программа заполняет все доступное пространство на диске, так что создание нового файла будет невозможно, то система будет доступна, но она не будет готова к выполнению задач.

В Trusted AIX предусмотрены механизмы защиты системы от несанкционированных действий, которые могут привести к неготовности системы. Непривилегированным процессам не разрешается читать защищенные файлы и каталоги и писать в них.

Отчетность

Это понятие относится к предотвращению угроз, связанных с незнанием того, какие именно процессы выполняют действия в системе. Например, если не удается определить, какой именно пользователь или процесс изменил системный файл, то невозможно будет найти способ предотвращения таких действий в будущем.

Эта расширенная функция защиты требует идентификации и проверки подлинности всех пользователей в системе, прежде чем им будут предоставлены какие-либо права доступа. Службы контроля ведут учет событий и заносят все события, связанные с защитой, в контрольный журнал.

Свойства Trusted AIX

- Trusted AIX устанавливается из меню установки AIX. Во время установки Trusted AIX можно указать дополнительные опции.
- Для восстановления обычной среды AIX после установки Trusted AIX потребуется выполнить установку AIX с заменой всех данных.
- Действия root не регистрируются в протоколе в среде Trusted AIX.
- В Trusted AIX все созданные WPAR также будут работать в среде с метками защиты.
- Trusted AIX поддерживает как MAC (обязательный контроль доступа), так и MIC (обязательный контроль целостности). Можно определить разные наборы меток для MAC и MIC.
- Файл LabelEncodings расположен в каталоге `/etc/security/enc` и содержит информацию для преобразования меток в двоичный формат и обратно. Файл LabelEncodings по умолчанию следует соглашению об именах меток Compartmented Mode Workstations (CMW).
- Установка NIM поддерживается при ее выборе в клиенте. Установка NIM методом рассылки с сервера невозможна, так как вход от имени root в системы MLS запрещен.
- Файловая система JFS2 (J2) (с расширенными атрибутами версии 2) поддерживает сохранение меток в AIX. Прочие файловые системы (J1 или NFS) могут быть смонтированы в Trusted AIX только как одноуровневые файловые системы (метка присваивается точке монтирования).
- Среда X выключена в Trusted AIX.
- Trusted AIX поддерживает протоколы CIPSO и RIPSO для обмена данными в сети на основе меток. Эти протоколы поддерживаются и для IPv4, и для IPv6.
- Некоторые механизмы защиты AIX являются общими для обычного AIX и Trusted AIX. Два таких механизма - это ролевое управление доступом (RBAC) и защищенное выполнение для проверки целостности.
- Так как при установке Trusted AIX пользователь root выключается, программа установки должна указать пароли пользователей ISSO, SA и SO в ходе первой загрузки после установке. Система не доступна, пока эти пароли не будут созданы.
- В технической публикации по функциям защиты в AIX 6 приведены варианты использования и примеры работы с Trusted AIX.

Многоуровневая защита

Главная цель защищенной системы - обеспечить учет и готовность за счет применения стратегии защиты сайта.

Стратегия защиты Trusted AIX предоставляет стандартный набор правил, определяющих типы разрешенного доступа. Сюда входят обеспечение подотчетности пользователей за их действия и предотвращение изменений в операционной системе.

Trusted AIX применяет контроль доступа и конкретные критерии "необходимо знать" для управления доступом к файлам, каталогам, процессам и устройствам.

Trusted AIX ведет журнал контроля всех событий, относящихся к защите. Этот журнал контроля позволяет вести учет на уровне отдельных пользователей, даже в программах, изменяющих действующие и фактические ИД пользователей, таких как команда **su**. Trusted AIX также ограничивает доступ к административным функциям, предоставляя его только конкретным пользователям с соответствующими правами доступа и наименьшими привилегиями (это наиболее ограничительный набор привилегий, позволяющих пользователю или процессу выполнить операцию).

Идентификация и проверка подлинности

Механизмы идентификации и проверки подлинности (I&A) отвечают за то, чтобы всякий пользователь, запрашивающий доступ к системе, был правильно идентифицирован и удостоверен. Для идентификации необходимо имя пользователя, а для проверки подлинности - пароль.

Все учетные записи Trusted AIX защищены паролем. Пользователь с правами ISSO (Information Systems Security Officer) может настроить систему так, что всякий пользователь сможет сам выбирать свой пароль, с учетом требований к длине и сложности пароля. Кроме того, пользователь с правами ISSO может указать параметры минимального и максимального возраста пароля (сроки действия пароля) для каждого пользователя, включая периоды выдачи предупреждений о предстоящем истечении срока действия пароля.

Механизмы идентификации и проверки подлинности требуют, чтобы все имена пользователей и ИД пользователей были уникальными. Учетные записи без допустимых паролей нельзя использовать для входа в систему. Пользователь с ролью ISSO должен добавить начальный пароль для всех новых пользователей. Каждому пользователю присваивается дополнительный уникальный идентификатор, применяемый для контроля.

Пароли хранятся только в зашифрованном виде. В обычном виде пароли не хранятся в системе. Зашифрованные пароли хранятся в теневом файле паролей, доступном только привилегированным процессам. Дополнительная информация приведена в описании команды **passwd**.

Системы Trusted AIX распознают учетные записи двух типов: системные и пользовательские. Системными считаются учетные записи с ИД пользователя меньше 128. Хотя с системными учетными записями могут быть связаны пароли, их нельзя использовать для входа в систему.

Избирательный контроль доступа

Параметры избирательного контроля доступа (DAC) - это параметры защиты, контролируемые владельцем файла или каталога.

Права доступа UNIX

Пользователь с правами владельца ресурса может выполнять следующие действия:

- Непосредственно предоставлять доступ другим пользователям
- Предоставлять доступ к копии другим пользователям
- Предоставлять программу, разрешающую доступ к исходному ресурсу (например, с помощью программ SUID)

Примером DAC может служить традиционный способ предоставления доступа с помощью разрешающих битов UNIX (владелец/группа/прочие и чтение/запись/выполнение).

Разрешающие биты дают возможность предоставлять или запрещать доступ к содержимому файла пользователям и группам (на основе критерия "необходимо знать"). Этот тип доступа основан на ИД пользователя и группах, в которые входит пользователь. Со всеми объектами файловой системы связаны определенные права доступа, описывающие доступ владельца, группы и прочих пользователей.

Владелец файла может также предоставить права доступа другим пользователям, изменив принадлежность или группу файла с помощью команд **chown** и **chgrp**

umask

При создании файла все разрешающие биты изначально включены. Затем некоторые из них отключаются процессом umask, который был настроен при входе в систему. По умолчанию umask применяется к каждому файлу, создаваемому оболочкой пользователя, и каждой команде, запускаемой из оболочки пользователя.

По умолчанию значение umask для элементов ядра равно 000 (что оставляет все права доступа всем пользователям). AIX устанавливает umask ядра 022 (что отключает биты записи для группы и прочих пользователей). Однако при необходимости пользователи могут переопределить это значение.

Примечание: Будьте очень осторожны при изменении umask на более разрешительное по сравнению с 022 значение. При расширении прав доступа к файлам и процессам система в целом становится менее защищенной.
Существует два способа переопределить значение umask по умолчанию:

- Вы можете изменить значения umask в своих файлах `.profile`, `.login` или `.chsrc`. Эти изменения повлияют на все файлы, создаваемые в вашем сеансе.
- Вы можете задать уровни umask для отдельных процессов с помощью команды **umask**. После выполнения команды **umask** новое значение umask будет влиять на все вновь создаваемые файлы, пока не произойдет одно из следующих двух событий:
  - Вы еще раз выполните команду **umask**

    ИЛИ
  - Вы завершите работу с оболочкой, в которой была выполнена команда **umask**

Если вы запустили команду **umask** без аргументов, то команда **umask** возвратит текущее значение umask для сеанса.

Вы должны разрешить своему сеансу наследовать значение umask 022 ядра. Для этого не указывайте umask в своих профайлах. Значения umask, менее ограничительные, чем 022, следует использовать с большой осторожностью.

Если для некоторых файлов необходимы дополнительные права доступа, то их следует задать с помощью команды **chmod** с соответствующими параметрами после создания файлов.

Списки управления доступом

Помимо стандартных разрешающих битов UNIX и значения umask, AIX поддерживает также списки управления доступом (ACL).

Разрешающие биты UNIX управляют доступом только таких категорий, как владелец файла, группа и прочие пользователи. Посредством ACL владелец файла может указать права доступа дополнительных конкретных пользователей и групп. Как и разрешающие биты, списки управления доступом связаны с отдельными системными объектами, такими как файл или каталог.

Разрешающие биты setuid и setgid

Разрешающие биты setuid и setgid (задать ИД пользователя и задать ИД группы) позволяют запускать файл программы с ИД пользователя или группы владельца файла, а не ИД пользователя или группы лица, выполняющего программу. Это достигается за счет задания битов setuid и setgid, связанных с файлом. Указанная возможность позволяет разрабатывать защищенные подсистемы, в которых пользователи могут получать доступ и запускать определенные файлы, не обязательно являясь их владельцами.

Если в родительском каталоге задан бит setgid при создании объекта, то новый объект будет входить в ту же группу, что и родительский каталог, а не в группу инициатора создания объекта. Однако если объект создается в каталоге с заданным битом setuid, то он будет принадлежать инициатору создания, а не владельцу каталога. При создании подкаталогов они наследуют биты setuid и setgid родительского каталога.

Разрешающие биты setuid и setgid представляют потенциальную угрозу безопасности. Программа, запускаемая с правами владельца root, получает практически неограниченный доступ к системе. Однако в системах Trusted AIX применение прав доступа и прочих параметров управления доступом существенно снижает эту угрозу безопасности.

Элементы управления ролевым доступом

Trusted AIX поддерживает Управление ролевым доступом (RBAC). RBAC - это механизм операционной системы, посредством которого специальные системные функции пользователя root или администратора могут выполняться обычными пользователями в соответствии с присвоенными им ролями.

Ниже перечислены базовые элементы RBAC AIX:

Права доступа
Эти строки указывают привилегированную операцию, которую они представляют и непосредственно контролируют по имени. Например, строка прав доступа `aix.network.manage` определяет функцию сетевого управления в AIX.

Привилегии
Привилегия - это атрибут процесса, позволяющий ему обойти системные ограничения и запреты. Привилегии связаны с процессом и обычно приобретаются путем выполнения привилегированной команды.

Роли Роли в RBAC AIX позволяют пользователям комбинировать набор управляющих функций в системе и предоставлять эти функции обычным пользователям. Роли в AIX состоят из прав доступа (это могут быть как системные, так и нестандартные права доступа) и других ролей (субролей).

Дополнительная информация об управлении ролевым доступом приведена в описании RBAC.

Обязательный контроль доступа

Обязательный контроль доступа (MAC) - это системный способ ограничения доступа к объектам в зависимости от секретности объекта и допуска пользователя. Напротив, избирательный контроль доступа применяется отдельными владельцами файлов, а не системой.

Применение меток для MAC

Применение MAC в Trusted AIX основано на системе меток. В системе Trusted AIX у всех именованных объектов есть метки секретности (SL), определяющие уровень секретности объекта. У процессов также есть SL. SL процессов указывают, какие уровни секретной информации доступны процессам. В общем случае, объект доступен процессу, если уровень секретности процесса больше или равен уровню секретности объекта. С помощью SL можно делать файлы доступными только для чтения или полностью недоступными обычным пользователям.

У всех системных объектов - файлов, объектов IPC, сетевых соединений и процессов - есть SL. SL автоматически присваиваются объектам при их создании. Все дампы ядра считаются объектами и автоматически помечаются системой.

Объекты, существующие до установки Trusted AIX, получают SL по умолчанию, `SYSTEM_LOW` (SLSL), при обращении к этим объектам после установки Trusted AIX. SL этих объектов не постоянны. Их можно изменить с помощью команды **settxattr**. Объектам, созданным после установки Trusted AIX, присваивается SL создавшего их процесса.

Пользователи и метки

Система присваивает каждой учетной записи пользователя диапазон допустимых SL, либо по системному значению по умолчанию, либо по пользовательскому параметру, и пользователь может работать только в пределах этого диапазона. Процесс или пользователь может создавать файлы и каталоги только на текущем уровне секретности процесса или пользователя и считывать и записывать файлы только в соответствии с налагаемыми системой ограничениями MAC.

Применение MAC

Обязательный контроль доступа применяется каждый раз, когда процесс пытается открыть объект файловой системы, извлечь атрибуты объекта файловой системы, отправить сигнал процессу, передать данные через STREAM или отправить или принять пакет через сетевой интерфейс. Доступ к любому объекту файловой системы возможен только при одновременном соблюдении критериев MAC и DAC. Когда пользователь пытается обратиться к файлу, сначала проверяются ограничения MAC, а затем - ограничения DAC, такие как разрешающие биты или списки управления доступом.

Доступ к объектам файловой системы ограничивается не только SL объекта, но и SL каталога объекта. Таким образом, объект файловой системы можно защищать на разных уровнях секретности - на уровне объекта и на уровне каталога. У объекта файловой системы может быть несколько имен (ссылок), расположенных в одном или нескольких каталогах. Хотя каждое имя (ссылка) защищено на том же уровне, что и файл, на который указывает ссылка, фактическая защита различных ссылок может быть разной, поскольку ссылки находятся в каталогах с различными уровнями защиты.

Имя объекта хранится в каталоге объекта. Таким образом, любой процесс, которому доступен этот каталог, может просматривать имена всех объектов каталога. Однако считывать из объектов и записывать в них могут только процессы с соответствующими привилегиями.

Просмотр и изменение SL

SL объектов и процессов в системе можно просмотреть командой **lstxattr** и изменить командой **settxattr**.

Изменить SL файла или процесса могут только пользователи с соответствующими правами доступа и процессы с соответствующими привилегиями.

Для изменения SL объекта файловой системы на более низкое значение с помощью команды **settxattr** у пользователя должны быть права доступа `aix.mls.label.sl.downgrade`. Для обновления SL объекта файловой системы у пользователя должны быть права доступа `aix.mls.label.sl.upgrade`. Для изменения SL процессов у пользователя должны быть права `aix.mls.proc.sl.upgrade` для повышения SL и права `aix.mls.proc.sl.downgrade` - для понижения SL.

MAC в дескрипторах открытых файлов

Когда процесс обращается к файлу для чтения, записи или выполнения простых операций, происходит проверка MAC. Если у процесса есть дескриптор этого файла, то он может считывать и записывать файл, даже если SL процесса изменилась на более низкую по сравнению с SL файла. Однако некоторые операции, например задание владельца файла, прав доступа, меток и привилегий, выполняют проверку доступа после того, как процесс получит дескриптор файла.

Это означает, что проверки MAC и преобразования пути к разделенному каталогу не выполняются, когда процесс обращается к файлу с помощью дескриптора файла. SL файла и/или процесса может измениться, но доступ по-прежнему будет разрешен.

Обязательный контроль целостности

Обязательный контроль целостности (MIC) - это системный способ ограничения доступа и изменения объектов в зависимости от целостности объекта и допуска пользователя. В то время как MAC контролирует уровень секретности объекта, MIC отвечает за его надежность.

Применение меток для MIC

Применение MAC в Trusted AIX основано на системе меток. В системе Trusted AIX у всех именованных объектов есть метки целостности (TL), определяющие уровень целостности объекта. У процессов также есть TL. TL процессов указывают уровень целостности информации, доступный процессу. Чем выше TL, тем более надежен объект или процесс.

Процесс может изменить объект только при условии, что его надежность не меньше, чем у объекта. Таким образом, TL процесса должна быть больше или равна TL объекта. Следовательно, метки целостности позволяют сделать файлы доступными только для чтения.

Кроме того, процесс не может использовать данные из объекта, менее надежного, чем сам процесс. Таким образом, в этом случае TL объекта должна быть больше или равна TL процесса.

У всех системных объектов, таких как файлы или процессы, есть TL. TL автоматически присваиваются объектам при их создании. Все дампы ядра считаются объектами и автоматически помечаются системой.

Объекты, существующие в системе до установки Trusted AIX, получают TL по умолчанию, `SYSTEM_LOW` (SLTL), при обращении к этим объектам после установки Trusted AIX. TL этих объектов не постоянны. Их можно изменить с помощью команды **settxattr**. Объектам, созданным после установки Trusted AIX, присваивается уровень целостности создавшего их процесса.

Пользователи и метки

Система присваивает каждой учетной записи пользователя диапазон допустимых TL, либо по системному значению по умолчанию, либо по пользовательскому параметру, и пользователь может работать только в пределах этого диапазона. Процесс или пользователь может создавать файлы и каталоги только на текущем уровне целостности процесса или пользователя и считывать и записывать файлы только в соответствии с налагаемыми системой ограничениями MIC.

Применение MIC

Обязательный контроль целостности применяется тогда же, когда и MAC. Кроме того, MIC применяется при удалении или переименовании файла или каталога.

Изменение TL

TL объектов и процессов можно просмотреть командой **lstxattr** и изменить командой **settxattr**.

Изменить TL файла или процесса могут только пользователи с соответствующими правами доступа и процессы с соответствующими привилегиями. Для изменения TL объекта файловой системы на более низкое значение с помощью команды **settxattr** у пользователя должны быть права доступа `aix.mls.label.tl.downgrade`. Для обновления TL объекта файловой системы у пользователя должны быть права доступа `aix.mls.label.tl.upgrade`. Для изменения TL процессов у пользователя должны быть права `aix.mls.proc.tl.upgrade` для повышения TL и права `aix.mls.proc.tl.downgrade` - для понижения TL.

NOTL

Существует специальная TL, `NOTL`, применимая к файловым системам, объектам IPC и процессам. Когда объекту или процессу присвоена TL `NOTL`, проверки MIC для объекта или процесса не выполняются. Задать TL равной `NOTL` или изменить TL на `NOTL` могут только привилегированные пользователи.

MIC в дескрипторах открытых файлов

Когда процесс обращается к файлу для чтения, записи или выполнения простых операций, происходит проверка MIC. Если у процесса есть дескриптор этого файла, то он может считывать и записывать файл, даже если TL процесса изменилась на более низкую по сравнению с TL файла. Однако некоторые операции, например задание владельца файла, прав доступа, меток и привилегий, выполняют проверку доступа после того, как процесс получит дескриптор файла. Это означает, что проверки MIC не выполняются, когда процесс обращается к файлу с помощью дескриптора файла. TL файла и/или процесса может измениться, но доступ по-прежнему будет разрешен.

Метки

Метки служат для представления уровней защиты для субъектов и объектов в системах Trusted AIX. Метки, применяемые в системе, и взаимосвязи между ними определяются пользователем с правами ISSO.

Метки секретности (SL):

SL, связанные со всеми субъектами и объектами, применяются для реализации стратегии управления доступом, основанной на модели Bell-LaPadula.

SL состоит из двух частей:

- Структурированная категория
- Набор из одного или нескольких отделов

Для каждой среды установки можно определить имена и взаимосвязи меток в системе. Системный администратор настраивает эти имена и взаимосвязи согласно требованиям стратегии среды в файле кодировок меток.

Категории SL:

Категории устроены иерархически и представляют уровень секретности.

Например, если на сайте допустимы категории `Top Secret`, `Secret` и `Unclassified`, то `Top Secret` более секретна, чем `Secret`, а `Secret` более секретна, чем `Unclassified`. Trusted AIX поддерживает до 32000 иерархических категорий.

Отделы SL:

Отделы представляют темы или рабочие группы. У каждого отдела есть имя, например `NATO` или `CRYPTO`.

У отделов нет внутренней упорядоченности, но пользователь с правами ISSO может наложить ограничения на возможные сочетания отделов и категорий. Trusted AIX поддерживает до 1024 отделов.

Компоненты SL:

SL - это строка, состоящая из нескольких элементов. Первым элементом является категория, остальные элементы - это отделы. Элементы разделены пробелами.

Например, для файла с секретной информацией о бразильской экономике категорией будет `TS` (совершенно секретно, top secret), а отделами могут быть Бразилия (`B`) и экономика (`e`). В удобочитаемом виде SL может быть представлена как `TS B e` или `Совершенно секретно Бразилия экономика`.

Отношения SL:

Пользователь системы должен понимать, как применяются метки и как они связаны между собой.

Существует три типа отношений между метками MAC:

- Поглощение
- Равнозначность
- Несопоставимость

Поглощение

Говорится, что одна SL (`L1`) поглощает другую (`L2`), если выполняются следующие два условия:

- Категория `L1` равна категории `L2` или превосходит ее

- Набор отделов L1 полностью содержит набор отделов L2

Например, рассмотрим метки SL L1 с совершенно секретной информацией с отделами A и B (TS A B) и SL L2 с секретной информацией с отделом A, но не с отделом B (S A). TS A B поглощает S A, так как категория TS поглощает категорию S, и набор отделов L1 полностью содержит набор отделов L2. L2 не поглощает L1 в этом примере.

Таблица 34. Поглощение SL

L1		L2		Поглощение
Метка	**Отдел**	**Метка**	**Отдел**	
Совершенно секретно	A,B	Секретно	A	L1 > L2

Равнозначность

Одна SL (L1) считается равнозначной другой SL (L2) только при выполнении следующих двух условий:

- Категория L1 равна категории L2
- Набор отделов L1 совпадает с набором отделов L2

Если метки равнозначны, то каждая из них поглощает другую. Например, метки SL для файла с совершенно секретной информацией с отделом A (TS A) и другого файла с секретной информацией с отделом A (тоже TS A) равнозначны и поглощают друг друга.

Таблица 35. Равнозначность SL

L1		L2		Поглощение
Метка	**Отдел**	**Метка**	**Отдел**	
Совершенно секретно	A	Совершенно секретно	A	L1 = L2

Несопоставимость

Две SL могут быть несопоставимы (L1 не равнозначна L2, L1 не поглощает L2, L2 не поглощает L1). Одна SL (L1) считается несопоставимой с другой SL (L2) только при выполнении следующего условия:

- Набор отделов L1 не содержит полностью набор L2, и набор L2 не содержит полностью набор L1. При этом L1 и L2 считаются несопоставимыми

Например, если есть файл с меткой L1 с совершенно секретной информацией и отделами A и B (TS A B) и файл с меткой L2 с конфиденциальной информацией и отделом C (C C), то L1 и L2 будут несопоставимыми.

Таблица 36. Несопоставимые SL

L1		L2		Поглощение
Метка	**Отдел**	**Метка**	**Отдел**	
Совершенно секретно	A, B	Конфиденциально	C	-

Метки целостности (TL):

TL представляют уровень надежности системного объекта или процесса. Структура TL та же, что и у SL, за исключением того, что у TL есть только иерархические категории и нет отделов.

Процесс может изменить или удалить объект, только если TL процесса поглощает TL объекта. Процесс может удалить или переименовать объект, только если TL процесса поглощает и TL объекта, и TL каталога объекта. Процесс может получить доступ к объекту, только если TL объекта поглощает TL процесса.

Для определения TL объекта или процесса воспользуйтесь командой **lstxattr**. Для изменения TL объекта или процесса - командой **settxattr**.

Метки субъектов и объектов:

В Trusted AIX процессы считаются субъектами и у каждого процесса есть метки секретности (SL).

SL, используемая для проверок MAC, называется действующей SL (ESL). ESL должна лежать в пределах диапазона допуска процесса. У диапазона допуска есть верхняя граница и нижняя граница. Верхняя граница называется Максимальным допуском (максимальным CL), а нижняя граница - Минимальным допуском (минимальным CL). ESL, максимальный CL и минимальный CL хранятся в структуре разрешений процесса и присваиваются во время создания процесса. Максимальный CL должен поглощать минимальный CL и ESL, а ESL должна поглощать минимальный CL. Просмотреть и задать SL процессов можно с помощью команд **settxattr** и **lstxattr**.

Доступ к различным объектам в системе требуется контролировать. Ниже перечислены возможные типы объектов:

- процесс
- файлы (файлы данных или двоичные файлы)
- объекты IPC, сетевые пакеты и т.п.

Все объекты и субъекты в системе MLS снабжаются метками.

Каталог
Каталоги связываются с диапазоном SL; минимальной SL и максимальной SL. Максимальная SL должна поглощать или быть равна минимальной SL. Все файлы каталога лежат в этом диапазоне.

Файлы Обычные файлы связаны с двумя SL, но их значения всегда совпадают. Фактически, у них только одна SL. У символьных ссылок могут быть разные значения SL.

Специальные файлы
Специальные файлы - устройства, терминалы и устройства типа FIFO - связаны с максимальной и минимальной SL. У каталогов, файлов и специальных файлов только одна метка целостности (TL), в то время как процессы связаны с минимальной и максимальной TL.

Процесс
Все процессы связаны с диапазоном, определяемым максимальным и минимальным допусками секретности, и диапазоном, определяемым максимальным и минимальным допусками целостности. Эти значения наследуются из значений допусков пользователя. Уровни секретности и целостности, на которых выполняется процесс, называются действующими уровнями секретности и целостности.

Метки допусков пользователей:

У пользователей есть максимальная и минимальная метки допусков секретности (SCL) и максимальная и минимальная метки допусков целостности (TCL)

Максимальная и минимальная метки допусков секретности

У каждого пользователя есть максимальная метка допуска секретности (максимальная SCL). Действующая SL пользователя должна поглощаться максимальной SCL. Максимальная SCL служит для запрещения конкретным пользователям просматривать секретные данные. Минимальная SCL служит для запрещения пользователям с высоким уровнем защиты передавать данные пользователям с более низким уровнем защиты.

Предположим, например, что у пользователя `A` максимальная SCL и минимальная SCL обе равны `PUBLIC_A`, а у пользователя `B` максимальная SCL и минимальная SCL обе равны `PUBLIC_B`. При отсутствии минимальной SCL пользователь `A` мог бы передавать информацию пользователю `B` путем входа в систему с действующей SL `IMPL_LO` и записи в файл, который пользователь `B` мог бы потом прочесть. При наличии минимальной SCL, однако, пользователь `A` должен войти в систему с `PUBLIC_A` и может записывать файлы

только на уровне PUBLIC_A. Любые файлы, записанные на уровне PUBLIC_A, недоступны для чтения пользователю B.

Максимальная и минимальная метки допусков целостности

У каждого пользователя есть также максимальная метка допуска целостности (максимальная TCL). Действующий TL пользователя должен поглощаться максимальной TCL. Максимальная TCL служит для запрещения конкретным пользователям просматривать секретные данные. Минимальная TCL служит для запрещения пользователям с высоким уровнем защиты передавать данные пользователям с более низким уровнем защиты.

Метки объектов файловых систем:

Каждый файл содержит конкретную информацию о защите. При создании нового файла ему присваивается SL процесса, создавшего файл. SL информации в файле можно повысить или понизить путем соответствующего изменения SL файла.

При создании каталога ему присваиваются минимальная SL и максимальная SL. При создании каталога обоим этим параметрам присваивается действующая SL создающего процесса, в результате чего фактически создается одноуровневый каталог. Изменять эти SL могут только пользователи с соответствующими привилегиями и правами доступа. Создавать новые объекты в этом каталоге можно только в том случае, если действующая SL процесса, создающего новый объект, попадает в диапазон SL каталога.

Окно обычно создается как отдельный дочерний процесс с SL, равной действующей SL пользователя. С устройствами (например, псевдотерминалами, связанными с окнами) также связаны SL. Именованный канал, который является устройством, применяемым для взаимодействия между процессами, наследует действующую SL процесса, создавшего этот канал. Поток, который является устройством, применяемым для предоставления двунаправленного канала данных для взаимодействия между процессами, также наследует действующую SL процесса, создавшего канал.

У всех устройств есть минимальная SL и максимальная SL. Максимальная SL должна поглощать минимальную SL. По умолчанию минимальная и максимальная SL задаются равными. Процесс может получить доступ к такому устройству в режиме чтения, только если SL процесса поглощает минимальную SL устройства или каталога. Процесс может получить доступ к такому устройству в режиме записи, только если SL процесса попадает в диапазон, определенный минимальной и максимальной SL устройства или каталога.

Флаги защиты файлов

Объекты могут помечаться флагами защиты файлов (FSF), влияющими на способ взаимодействия процессов с объектами. Список FSF и привилегий, необходимых для задания каждого FSF, приведен в разделе Флаги защиты файлов. У процессов нет флагов защиты файлов.

Удаление файлов:

Удалить объект из файловой системы можно только при соблюдении следующих условий:

- Процесс, пытающийся удалить объект, должен видеть имя файла в каталоге файла. Это означает, что у процесса должны быть права на поиск к каждому каталогу в пути к каталогу удаляемого объекта, а также действующая SL, поглощающая каждый из этих каталогов. Для просмотра имени файла служит команда **ls**.
- У процесса должны быть права на запись к каталогу удаляемого объекта.

Печать файлов:

Подсистема принтера автоматически помечает весь вывод соответствующими метками секретности. Для каждого задания печати автоматически создаются начальная и конечная страницы, содержащие все относящиеся к защите метки и указатели.

Резервное копирование и восстановление файлов:

При записи данных на диски или магнитные ленты в AIX командой **backup** SL записываются вместе с данными.

Для импорта или экспорта данных без меток с магнитных лент или дисков командами **backup** или **restore** необходимы права доступа SO. При записи данных без меток присваивается SL по умолчанию `SYSTEM_LOW` для файлов и диапазон SL с `SYSTEM_LOW` по `SYSTEM_HIGH` для каталогов.

Метки в объектах IPC:

Все компоненты IPC AIX предусматривают создание и доступ к промежуточным объектам.

В AIX определены три различных компонента IPC:
- Очереди сообщений
- Семафоры
- Общая память

Все они предусматривают создание и доступ к промежуточным объектам, называемым объектами IPC, для обмена информацией между процессами. Каждый объект IPC защищен набором атрибутов, схожих с атрибутами защиты файлов. Это следующие атрибуты:
- ИД пользователя и ИД группы владельца объекта
- ИД пользователя и ИД группы создателя объекта
- Режим доступа к ресурсам, аналогичный разрешающим битам доступа к файлам. Для каждого объекта определены права на чтение, запись и выполнение для владельца объекта, группы и прочих пользователей.
- Порядковый номер для отслеживания использования ресурсов
- Ключ для идентификации ресурса

Как и в случае остальных системных объектов, Trusted AIX добавляет к этим атрибутам новые атрибуты защиты. В системе Trusted AIX у всех объектов IPC есть также следующие атрибуты:
- Метка секретности (SL)
- Метка целостности (TL)

Просмотреть все атрибуты защиты объекта IPC можно командой **settxattr**. Для чтения атрибутов объекта IPC необходимы права доступа `DAC READ` и `MAC READ` к объекту.

Доступ к объектам IPC:

Создание, удаление объектов IPC, а также доступ к ним осуществляются посредством нескольких системных вызовов, описанных в главе Программирование Trusted AIX. Эти операции не выполняются обычными пользователями. В этом разделе представлен общий обзор правил создания и удаления объектов IPC, а также доступа к ним.

Для получения доступа к объекту IPC процесс должен пройти проверки DAC, MIC и MAC.

В основе проверок доступа DAC лежит режим объекта (владелец, группа или глобальный) объекта и ИД пользователя и группы процесса. Процесс получает доступ DAC в качестве владельца к объекту IPC, если действующий UID процесса совпадает с UID владельца или автора объекта. Этот же принцип применяется и к групповому доступу DAC.

Доступ MAC основан на SL процесса и объекта. Доступ MIC основан на TL процесса и объекта.

Правила доступа к содержимому объекта IPC такие же, как и для атрибутов объекта IPC. Для прочтения содержимого или атрибутов объекта IPC требуются права доступа `DAC READ`, `MIC READ` и `MAC READ`. Для записи в объект IPC требуются права доступа `DAC WRITE`, `MIC WRITE` и `MAC WRITE`.

Атрибуты объекта IPC подчиняются более жестким ограничениям по сравнению с содержимым объекта IPC. Следовательно, для изменения атрибутов объекта IPC необходимы более широкие права доступа. Для изменения стандартных атрибутов AIX, таких как режим, процессу должны быть присвоены права доступа `DAC OWNER` и `MAC WRITE`. Для того, чтобы изменить SL объекта IPC, процессу должны быть присвоены все перечисленные ниже права доступа:

- `PV_SL_PROC`
- `DAC OWNER` (только понижение уровня)
- `DAC WRITE`
- `MAC WRITE`
- Права доступа `PV_SL_UG` для обновления SL или права доступа `PV_SL_DG` для понижения уровня SL
- п`PV_MAC_CL`, если они существуют, или новый SL за пределами допуска процесса
- `MIC WRITE`

Для того, чтобы изменить TL объекта IPC, процессу должны быть присвоены все перечисленные ниже права доступа:

- права доступа `PV_TL`
- `DAC OWNER`
- `MAC WRITE`
- `MIC WRITE`

Кроме того, для того, чтобы заблокировать или разблокировать сегмент общей памяти, процессу необходимы права доступа `PV_KER_IPC_O`. Кроме того, для изменения параметра очереди сообщений `msg qbytes` в процедуре $_{msgctl}$ процессу также должны быть присвоены права доступа `PV_KER_IPC`.

Понятия, связанные с данным:

“Программирование Trusted AIX” на стр. 462
Защита системы системы зависит от таких компонентов защищенной компьютерной базы (TCB), как программное обеспечение, аппаратное обеспечение и встроенное ПО. В это входит полностью ядро операционной системы, все драйверы устройств и модули System V STREAMS, расширения ядра и все защищенные программы. Все файлы, к которым обращаются эти программы при принятии решений, связанных с защитой, также являются частью TCB.

Создание и удаление объектов IPC:

На создание объектов IPC нет ограничений. Когда процесс создает объект IPC, объект наследует SL и TL процесса.

Режим доступа объекта IPC должен быть указан системным вызовом, создающим объект.

Для удаления объекта IPC у процесса должны быть права доступа `DAC OWNER`, `MIC WRITE` и `MAC WRITE` к объекту.

Защищенные сети:

Для улучшенной защиты систем предусмотрен ряд требований к защищенным сетям, относящийся к расширенным атрибутам защиты. Защищенная сеть AIX поддерживает несколько признанных стандартов защищенной сети, таких как RIPSO и CIPSO.

В AIX реализована поддержка защищенной сети как для IPv4, так и для IPv6. При связи с другими защищенными системами SL упаковывается в опции IP согласно стандартам CIPSO/RIPSO. Проверки MAC также осуществляются на уровне IP для SL, которые отправляются или принимаются с пакетами. Для разрешенного диапазона меток настраиваются правила сети. Правила сети включают в себя правила для хостов и правила для интерфейсов. Защищенная сеть AIX устанавливает только правила для интерфейсов по умолчанию (одно правило на настроенный интерфейс). Правила фильтрации можно уточнить с помощью правил для хостов. Настроить правила и для интерфейсов, и для хостов можно командой `netrule`. Команда `netrule` поддерживает такие операции, как добавление, удаление, показ и опрос правил.

Также можно использовать команду `tninit` для инициализации подсистемы защищенной сети и работы с базой данных правил защищенной сети.

Отключение root:

Учетная запись пользователя root отключена в системах Trusted AIX. Это сделано прежде всего с целью минимизировать ущерб, который может нанести системе отдельный пользователь, обладающий всеми привилегиями.

Отключены все типы входа в систему под именем root. Лишь команда **su** разрешает вход в систему под именем root. Процессам, принадлежащим root, не назначаются никакие специальные привилегии. Принадлежащие пользователю root программы setuid и non-setuid работают как и раньше, будучи запущенными пользователями с соответствующими правами доступа. Если у пользователя нет необходимых прав доступа, то программа будет запущена, если биты режима DAC или списки управления доступом разрешают выполнение, но программе не будут присвоены никакие привилегии, поэтому она не сможет выполнять привилегированные операции в этом случае. Таким образом, вновь устанавливаемым приложениям необходимо присваивать соответствующие привилегии, если предполагается, что эти приложения будут выполнять привилегированные операции.

Задачи системного администрирования могут выполнять пользователи с ролями Information System Security Officer (ISSO), System Administrator (SA) или System Officer (SO). Эти роли позволяют любому пользователю выполнять задачи системного администрирования.

Примечание: Во время установки Trusted AIX атрибуту **su** учетной записи root присваивается значение `false`. Для того чтобы разрешить доступ к учетной записи root другим административным пользователям, пользователь с правами ISSO должен сбросить этот атрибут к значению `true` командой **chuser** и присвоить пароль этой учетной записи.

Поддержка меток в контроле:

Основное предназначение подсистемы контроля - отслеживание и запись событий, относящихся к защите.

Информация, предоставляемая подсистемой контроля, позволяет записывать информацию следующих типов:

- Попытки нарушения стратегии защиты
- Успешное выполнение действий, относящихся к защите

Подсистема контроля позволяет:

- Определять, какие события следует контролировать
- Включать и выключать контроль во время работы системы

- Автоматически (без потери информации) переключать файлы журналов контроля
- Преобразовывать информацию контроля в удобочитаемую форму
- Выбирать и обрабатывать подмножества информации контроля

Во время настройки подсистемы контроля пользователь с правами ISSO должен понимать, что требуется контролировать, в каких условиях будет происходить контролирование и как запустить и завершить контролирование. Подробная информация о настройке, запуске и завершении, администрировании и просмотре контроля приведена в разделе Обзор контроля.

Подсистема контроля поддерживает свое текущее состояние и автоматически перезапускается с переходом в это состояние после отключения питания, сбоя системы, сбоя питания или другого прерывания. Подсистема контроля может автоматически завершить работу, завершить работу системы или изменить файлы контроля при возникновении ситуации, в которой дальнейшее хранение записей контроля в существующем файле контроля становится невозможным. Файлы контроля могут автоматически переключаться, когда файловая система переходит на указанный уровень. Однако в случае катастрофического сбоя питания возможна утеря небольшого числа записей контроля.

Многоуровневые и разделенные каталоги:

Многоуровневый каталог - это стандартный каталог, которому присвоена не отдельная SL, а диапазон SL. Разделенный каталог выглядит как обычный каталог для пользователя. Однако файлы, которые видит пользователь, на самом деле находятся в скрытом подкаталоге разделенного каталога.

Многоуровневые каталоги:

Многоуровневый каталог - это стандартный каталог, которому присвоена не отдельная SL, а диапазон SL.

Для просмотра имен файлов в многоуровневом каталоге процесс должен работать на уровне защиты, превышающем минимальную SL каталога. Для создания или удаления фактических файлов процесс должен работать в диапазоне SL многоуровневого каталога.

У каждого файла в многоуровневом каталоге есть своя собственная SL и файл защищен стандартными ограничениями MAC. Однако любой процесс, которому доступен этот каталог, может просматривать имена всех объектов каталога. Таким образом, возможны ситуации, когда у процесса есть права MAC на чтение каталога и запись в каталог, но процесс не может считывать или записывать некоторые файлы каталога, хотя и может просматривать имена всех файлов каталога.

Разделенные каталоги:

Разделенный каталог выглядит как обычный каталог для пользователя. Однако файлы, которые видит пользователь, на самом деле находятся в скрытом подкаталоге разделенного каталога.

Многоуровневые каталоги представляют угрозу безопасности. Процесс, работающий с высоким уровнем защиты, может прочесть файл с низким уровнем защиты и затем создать файлы на своем, высоком уровне защиты. Хотя компоненты MAC не позволяют процессам с низким уровнем защиты считывать новые файлы, таким процессам видны имена новых файлов. Если процесс с высоким уровнем защиты присвоил новым файлам имена в соответствии с содержимым исходного файла с высоким уровнем защиты, то процессы с низким уровнем защиты могут ознакомиться с информацией, защищенной на более высоком уровне, прочитав имена новых файлов.

Если создан разделенный каталог и процесс обращается к нему, то система создает скрытый подкаталог с SL этого процесса. Если затем процесс создает файл, то на самом деле файл создается в скрытом подкаталоге. В разделенном каталоге может быть несколько таких скрытых подкаталогов, но процесс, обращающийся к разделенному каталогу, будет видеть только те файлы, SL которых совпадает с SL процесса. Когда процесс создает дочерний каталог разделенного подкаталога, этот дочерний каталог будет разделенным подподкаталогом.

Разделенному каталогу присваивается диапазон SL от SYSTEM_LOW до SYSTEM_HIGH. Таким образом, разделенные каталоги доступны всем процессам.

Пользователи с правами доступа **aix.mls.pdir.mkdir** могут создавать разделенные каталоги командой **pdmkdir**. Пустые разделенные каталоги можно удалять командой **pdrmdir**. Команда **pdset** изменяет обычный каталог на разделенный. Команда, изменяющая разделенный каталог на обычный, не предусмотрена.

В разделенном каталоге файл, находящийся в одном разделенном подкаталоге, можно связать со всеми остальными разделенными подкаталогами с более высокими SL, расположенными в том же разделенном каталоге. В результате этого файл станет доступным всем процессам, которым доступен этот разделенный подкаталог или разделенные подкаталоги более высокого уровня в этом разделенном каталоге. Такое связывание файла можно выполнить командой **pdlink**.

Режимы доступа к разделенным каталогам:

При создании процессу присваивается один из двух режимов - реальный или виртуальный. Режим определяет, каким образом процесс просматривает разделенные каталоги.

Реальный процесс рассматривает разделенные каталоги как стандартные многоуровневые каталоги. Все разделенные подкаталоги доступны как стандартные каталоги, с учетом обычных ограничений DAC, MIC и MAC. Реальный процесс может войти в разделенный каталог и просмотреть все подкаталоги, с учетом ограничений DAC, MIC и MAC.

Виртуальный процесс никогда не входит в разделенный каталог, но перенаправляется в разделенный подкаталог, максимальная и минимальная SL которого обе равны действующей SL процесса.

Реальный процесс может запустить команду в виртуальном режиме с помощью команды **pdmode** (например, `pdmode ls`). Аналогично, виртуальный процесс может запустить команду в реальном режиме, также с помощью команды **pdmode** (например, `pdmode -r ls`). Однако для этого требуются права доступа `aix.mls.pdir.mode`. Кроме того, при наличии этих прав доступа вы можете переключиться из оболочки, работающей в виртуальном режиме, на оболочку, работающую в реальном режиме, выполнив `pdmode -r sh`. Для запуска программы в виртуальном режиме при работе в реальном режиме никаких прав доступа не требуется.

Просмотр и изменение типов каталогов:

Команда **lstxattr** позволяет показать тип каталога как часть атрибута `secflags`. `FSF_PDIR` обозначает разделенный каталог, `FSF_PSDIR` - разделенный подкаталог, а `FSF_PSSDIR` - разделенный под-подкаталог. Для того чтобы сделать обычный каталог разделенным каталогом, используйте команду **pdset**.

Администрирование Trusted AIX

Управление системой Trusted AIX обладает рядом особенностей, относящихся именно к Trusted AIX.

Установка Trusted AIX

Trusted AIX можно подключить только во время установки базовой операционной системы с помощью опции `Модель защиты` меню установки.

Опция переноса Trusted AIX не поддерживается. Для установки с сохранением необходимо применять файловую систему JFS2. В случае автономной сетевой установки обратитесь к таблице Табл. 37 на стр. 431, содержащей пароли административных пользователей по умолчанию.

Таблица 37. Пароли административных пользователей по умолчанию

Пользователь	Пароль
isso	isso
sa	sa
so	so

Режимы функционирования

Для настройки и обслуживания системы и для каждодневных операций предусмотрено два режима функционирования - режим настройки и рабочий режим.

Когда система загружается, она первоначально работает в режиме настройки. По окончании инициализации система переключается на рабочий режим.

Режим настройки применяется для обслуживания и восстановления системы. При загрузке системы в однопользовательском режиме выполняется минимальная настройка системы, а сеть отключается. Режим настройки используется для администрирования важнейших частей системы, связанных с ее защитой.

Рабочий режим - это стандартный режим работы системы. Система переходит в этот режим после выполнения всех задач, необходимых для переключения на уровень функционирования по умолчанию.

Режим работы системы можно просмотреть командой **getrunmode** и изменить командой **setrunmode**.

Флаги защиты ядра

Флаги защиты ядра служат для включения и отключения некоторых функций защиты, например проверки меток, проверки наличия меток целостности во время операций чтения и других.

Ядро проверяет флаги защиты ядра перед проведением проверок защиты. Эти флаги поддерживаются только при включенном Trusted AIX. В пользовательском пространстве эти флаги хранятся в базе данных ODM. Набор проверяемых ядром флагов защиты зависит от режима работы системы.

Таблица 38. Флаги защиты ядра и значения по умолчанию

Флаг защиты ядра	Включен	Отключен	Рабочий режим по умолчанию	Режим настройки по умолчанию
tnet_enabled	Функция надежной сети доступна	Функцию надежной сети нельзя ни настроить, ни использовать	Отключен	Отключен
tl_write_enforced	MIC применяется в операциях записи, удаления и переименования	Конфигурация задана так, что TL не используются в проверках записи	Включен	Включен
tl_read_enforced	MIC применяется в операциях чтения	Конфигурация задана так, что TL не используются в проверках чтения	Отключен	Отключен
sl_enforced	Применяется MAC	Конфигурация задана так, что SL не используются для управления доступом	Включен	Отключен
trustedlib_enabled	Флаг FSF_TLIB учитывается в объектах файловой системы	Флаги FSF_TLIB не учитываются	Отключен	Отключен

Задание параметров ядра

Ядро Trusted AIX можно настроить на обеспечение ограничений защиты, требуемых стратегиями организации.

Конфигурацию защиты можно просмотреть командой **getsecconf** и изменить командой **setsecconf**. Настраиваются следующие параметры ядра:

- Применение меток секретности
- Обеспечение целостности при чтении
- Обеспечение целостности при записи
- Защищенная сеть
- Защищенная библиотека

Эти параметры можно настроить только в режиме настройки системы.

Настройка файла /etc/security/enc/LabelEncodings

Системные метки задаются в файле `/etc/security/enc/LabelEncodings` и могут настраиваться для каждого сайта.

Метки можно настраивать после установки Trusted AIX.

В системе Trusted AIX определен `SYSTEM LOW SL` (`SLSL`), переопределяемый любой другой меткой секретности в системе, и `SYSTEM HIGH SL` (`SHSL`), переопределяющий любую метку секретности. Аналогично, `SYSTEM LOW TL` (`SLTL`) переопределяется любой другой меткой целостности в системе, а `SYSTEM HIGH TL` (`SHTL`) переопределяет любую метку целостности. Эти определения принимают значения наибольших и наименьших SL и TL, заданных в файле `/etc/security/enc/LabelEncodings`.

При загрузке системы Trusted AIX системные метки из файла `/etc/security/enc/LabelEncodings` загружаются в ядро. Загрузить метки в ядро можно также командой **setsyslab**. Просмотреть системные метки, определенные в ядре, можно командой **getsyslab**. После внесения изменений в файл `/etc/security/enc/LabelEncodings` рекомендуется перезагрузить систему.

Комментарии можно вставлять в любом месте файла `/etc/security/enc/LabelEncodings`, в котором может начинаться ключевое слово. Комментарии начинаются с `*` и продолжаются до конца строки.

Файл `/etc/security/enc/LabelEncodings` содержит информацию о версии и следующие обязательные разделы. Каждый раздел должен начинаться с одного из следующих ключевых слов, за которыми должно следовать двоеточие (`:`):

- classifications
- information labels
- sensitivity labels
- clearances
- channels
- printer banners
- accreditation range

Файл `/etc/security/enc/LabelEncodings` начинается с записи `VERSION`. Эта запись представляет собой последовательность символов и может содержать пробелы.

В разделе можно указывать следующие ключевые слова. Эти ключевые слова оканчиваются точкой с запятой (`;`):

name=*имя*
: Ключевое слово, определяющее полное имя категории или отдела

sname=*имя*
: Ключевое слово, определяющее сокращенное имя. Необязательно.

aname=*имя*
: Альтернативное ключевое слово для категории. Необязательно.

value=*значение*

Ключевое слово, задающее внутреннее целое значение категории или отдела

compartments=*бит*

Ключевое слово, указывающее, какой бит раздела должен быть равен 0 или 1, когда слово присутствует в метке

Усовершенствования формата кодирования меток в Trusted AIX

Кодирование меток, предписываемое документом Defense Intelligence Agency Document DDS-2600-6216-93, не поддерживает метки целостности.

По умолчанию в качестве меток целостности применяются метки секретности. Trusted AIX поддерживает необязательный раздел меток целостности, который может отличаться от разделов меток секретности. Это повышает гибкость системы, поскольку позволяет применять различные имена и значения категорий для меток секретности и целостности. Например, метки секретности можно снабдить префиксом SL, а метки целостности - префиксом TL, как указано ниже:

Таблица 39. Имена и значения категорий для меток секретности

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

Таблица 40. Имена и значения категорий для меток целостности

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

К разделу меток целостности применяются следующие правила:

- Раздел "INTEGRITY LABELS" должен следовать за разделом "NAME INFORMATION LABELS". Если администратор не определил необязательный раздел "NAME INFORMATION LABELS", то раздел "INTEGRITY LABELS" должен следовать за разделом "ACCREDITATION RANGE".
- В файле закодированных меток должен быть только один раздел "INTEGRITY LABELS". Он относится и к объектам, и к субъектам.
- Новый раздел "INTEGRITY LABELS" необязателен. Если этот раздел отсутствует, то следует применять категории, заданные в обязательном разделе "CLASSIFICATIONS".

- Раздел "INTEGRITY LABELS" схож с разделом "CLASSIFICATIONS". Он может содержать следующие ключевые слова: **"name="**, **"sname="**, **"aname="** и **"value="**. Ключевые слова **"initial compartments="** и **"initial markings="**, входящие в раздел "CLASSIFICATIONS", в разделе "INTEGRITY LABELS" недопустимы.
- Диапазон данных для параметра **"value="** совпадает с тем, который применяется в разделе "CLASSIFICATIONS": от минимального значения 0 до максимального значения 32000.

Запуск системы

Защита системы включается автоматически во время процедуры запуска системы. Во время запуска системы необходимо проверить правильность параметров защиты, которые показываются в ходе запуска.

Режим настройки:

Режим настройки применяется для обслуживания и восстановления системы.

При загрузке системы в однопользовательском режиме выполняется минимальная настройка системы, а сеть отключается.

Рабочий режим:

Рабочий режим применяется в каждодневной работе.

Как правило, систему следует загружать непосредственно в многопользовательском режиме. Если программа проверки прав доступа при загрузке получает допустимые имя пользователя и пароль, то система переходит в рабочий режим и появляется меню идентификации входа в систему с консоли, после чего пользователи могут входить в систему.

Механизмы защиты, например метки секретности, избирательный контроль доступа, обязательный контроль доступа, проверки привилегий, идентификация и проверка подлинности и проверка прав доступа, активны и в режиме настройки, и в рабочем режиме, согласно соответствующим флагам конфигурации защиты. Дополнительная информация приведена в описании команды **getsecconf**.

Все операции в системе рекомендуется выполнять в рабочем режиме, чтобы гарантировать доступность всех ожидаемых функций системы.

Процесс загрузки:

В файл `/etc/inittab` Trusted AIX систем добавлены новые сценарии загрузки. Новые сценарии загрузки - `rc.mls.boot`, `rc.mls.net` и `rc.mls`, и выполняются они в этом же порядке.

В сценарии `rc.mls.boot` выполняются следующие действия:

1. Выполняется интерактивная проверка целостности данных, чтобы пользователь получал информацию о том, как обрабатывать каждое отклонение (с помощью команды **trustchk**)
2. Установка флагов защиты ядра в режиме настройки (с помощью команды **setsecconf**)
3. Установка системных меток (метки минимальной и максимальной чувствительности и метки целостности данных)
4. Флаги защиты ядра в режиме настройки отображаются на экране

В сценарии `rc.mls.net` выполняются следующие действия:

1. Инициализация подсистемы Trusted AIX.
2. Если файл `/etc/security/rules.int` существует, он загружает в ядро базу данных правил.

В сценарии `rc.mls` выполняются следующие действия:

1. Инициализация подсистемы Trusted AIX.

2. Если файл /etc/security/rules.int существует, он загружает в ядро базу данных правил.

Примечание: Любое изменение сценариев загрузки может привести к сбою системы.

Настройка запуска системы:

Хотя это и не рекомендуется, и идентификацию, и проверку целостности системы, выполняемые при запуске, можно отключить.

За исключением случая, когда и идентификация, и проверка целостности системы отключены, запуск системы с консоли происходит с участием оператора.

Отключение идентификации BOOT:

Идентификацию BOOT можно отключить с помощью команды **rmitab bootauth** или меню SMIT.

Отключение проверки целостности системы:

Для отключения автоматической проверки целостности системы при загрузке удалите строку trustchk из сценария **rc.mls.boot**.

Завершение работы системы

Выключение системы - это привилегированная операция. Для ее выполнения необходимы права доступа aix.system.boot.shutdown.

Выключить систему может пользователь в роли SO или любой другой роли, которой предоставлены эти права доступа.

Восстановление системы Trusted

Иногда система выключается нештатным образом. Это может произойти из-за нарушения в системе электроснабжения, случайного выключения питания или отказа оборудования. Система Trusted AIX может восстановиться после этих ситуаций без применения особых процедур перезагрузки.

При перезагрузке системы включаются все механизмы обеспечения защиты, независимо от того, как была выключена система. В процедурах запуска проверяется, нет ли повреждений файловых систем. Сценарии запуска вызывают команду **fsck**, чтобы закрыть доступ неавторизованных пользователей к поврежденным или измененным файлам.

Команда **trustchk** сообщает о всех отклонениях в атрибутах защиты файлов и каталогов и предлагает пользователю исправить эти атрибуты. Команду **trustchk** необходимо вызывать всякий раз, когда возникает подозрение в нарушении целостности файловой системы. Дополнительная информация приведена в описании команды **trustchk**.

Вход в систему

У каждого пользователя Trusted AIX должны быть правильные допуски секретности и целостности, чтобы он мог входить в систему.

Допуски пользователя определяются как пользовательские атрибуты в файле /etc/security/user. Атрибуты minsl и maxsl определяют допуск секретности пользователя. Атрибуты mintl и maxtl - допуск целостности пользователя. Атрибуты defsl и deftl определяют действующие уровни секретности и целостности пользователя при входе в систему.

Атрибуты допуска пользователя можно изменить командами chuser и chsec и просмотреть командами lsuser и lssec.

Пользователи могут просматривать свои собственные метки, но не могут изменять их. Для просмотра меток допусков других пользователей необходимы права доступа `aix.mls.clear.read`. Для изменения допусков необходимы права доступа `aix.mls.clear.write`.

Для входа в систему необходимо, чтобы соблюдались все следующие правила поглощения:

- Значение `minsl` должно поглощаться значением `defsl`
- Значение `defsl` должно поглощаться значением `maxsl`
- Значение `mintl` должно поглощаться значением `deftl`
- Значение `deftl` должно поглощаться значением `maxtl`

Вы можете указать требуемые действующие уровни секретности и целостности во время входа в систему с помощью опций **–e** и **–t** команды **login**. Дополнительная информация приведена в описании команды **login**.

Для входа в систему на уровне секретности, не входящем в диапазон аккредитования системы, необходимы права доступа `aix.mls.label.outsideaccred`.

Trusted AIX не позволяет входить в систему системным пользователям (пользователям с ИД меньше 128).

Причины сбоев при входе в систему

Причины сбоев при входе в систему могут быть разными.

Попытка войти в систему окажется неудачной в следующих случаях:

- Введен недопустимый ИД входа в систему
- Введен недопустимый пароль
- Учетная запись помечена как заблокированная, поскольку число предыдущих неудачных попыток входа в систему для этой учетной записи превысило системное ограничение
- Порт входа в систему помечен как заблокированный, поскольку число предыдущих неудачных попыток входа в систему для этого порта превысило системное ограничение
- У ИД входа в систему нет допустимого допуска
- Указанная метка (или метка секретности или целостности по умолчанию для ИД входа в систему, если метка не указана) недопустима, не входит в допуск для ИД входа в систему, не входит в допуск устройства входа в систему или не входит в диапазон аккредитования системы
- У пользователя нет доступа DAC к пути программы оболочки входа в систему, либо у учетной записи пользователя нет доступа на исполнение DAC к программе оболочки входа в систему
- У пользователя нет доступа на чтение MAC или MIC к пути программы оболочки входа в систему или нет доступа на чтение MAC или MIC к программе оболочки входа в систему
- UID идентификатора входа в систему меньше 128

Смена пользователя командой su

В системе Trusted AIX команду **su** с опцией **-** может вызывать только пользователь, допуски которого шире допусков нового пользователя.

Для меток секретности и целостности должны выполняться следующие условия:

- максимальный допуск текущего пользователя должен поглощать максимальный допуск нового пользователя.
- минимальный допуск нового пользователя должен поглощать минимальный допуск текущего пользователя.
- фактический допуск текущего пользователя должен поглощаться максимальным допуском нового пользователя и должен поглощать минимальный допуск нового пользователя.

Ответственность пользователя за защиту

Пользователи должны знать и выполнять требования в отношении защиты. Пользователи не должны разглашать пароли, они должны отслеживать изменения в своем состоянии и сообщать о возможных угрозах безопасности.

Пароль

Пароли необходимо запоминать, а не записывать где бы то ни было. Если пароль станет известен другому пользователю, то это может быть угрозой защиты для всей системы.

Наиболее явной угрозой защиты системы является разглашение паролей. Регулярная смена паролей является простейшей мерой защиты от пользователей, тем или иным доступом узнавших чужой пароль. Пароли следует менять достаточно часто, чтобы за время его применения вероятность его несанкционированного использования была мала. Чем дольше используется пароль, тем больше вероятность его взлома.

Если пользователям разрешено выбирать собственные пароли, то длина пароля должна быть не менее шести символов, и он должен содержать как минимум две буквы и одну цифру. Пароль никак не должен быть связан с личными или профессиональными данными пользователя (например, с именем пользователя, его друзей, его собаки или с должностью), и он не должен быть обычным словом из словаря. В схемах подбора пароля часто используется словарь и набор личных данных, таких как имя пользователя, имена детей и домашних животных и дата рождения.

Срок действия паролей может установить ISSO. Если срок действия пароля истек, то при попытке входа в систему ему будет предложено сменить пароль, и войти в систему можно будет только с новым паролем. Рекомендуется изменять пароли чаще, чем в соответствии с их сроком действия. При любом подозрении на то, что пароль был подобран, его необходимо немедленно изменить.

Присутствие в системе

Ни в коем случае не следует оставлять возможность работать в сеансе пользователя во время его отсутствия. Если необходимо отойти от системы даже на короткое время, то рекомендуется завершить сеанс работы.

Управление защищенными системами

Управление защищенными компьютерными системами подразумевает создание и исполнение стратегии защиты и постоянный мониторинг системы.

Далее перечислены факторы, которые следует учесть при разработке стратегии защиты в организации:

- Максимальный уровень защиты в области системы не должен превышать максимальный уровень защиты во всей организации, в которой расположена система.
- Аппаратное обеспечение должно быть размещено в безопасном расположении. Наиболее безопасными обычно являются внутренние комнаты не на первом этаже.
- Физический доступ к системе должен быть ограничен и регламентирован, и должен отслеживаться.
- Носители резервных копий и архивов должны храниться в безопасном месте, отдельно от прочего оборудования сети. Физический доступ к этому расположению также должен быть ограничен, как и для системного аппаратного обеспечения.
- Доступ к руководствам и административной документации должен быть предоставлен только соответствующим сотрудникам, согласно необходимости.
- Перезагрузки системы, сбои питания и выключения должны записываться. Нарушения в файловой системе должны документироваться, и все поврежденные файлы должны быть проанализированы на предмет возможных нарушений стратегии защиты.
- Установка новых программ, полученных или созданных, должна быть ограничена и выполняться под контролем. Новые программы должны быть изучены и проверены перед их запуском.

- Непредвиденное поведение любой программы в системе должно быть задокументировано, и причина этого поведения должна быть определена.
- Если это возможно, то администрировать систему должны два сотрудника. Одному из них должна быть назначена роль `isso`, а другому - роль `sa`.
- Не следует использовать права доступа `PV_ROOT`. Для администрирования системы должно быть достаточно прав пользователей ISSO, SA или SO.
- Данные контроля должны регистрироваться в протоколе и регулярно просматриваться. Нештатные или необычные события должны отмечаться, и их причина должна быть определена.
- Число сеансов работы в ролях `isso`, `sa` и `so` должно быть минимальным.
- Число программ setuid и setgid должно быть минимальным, и их следует использовать только в защищенных подсистемах.
- Права доступа, предоставляемые новым программам, должны быть минимальными и согласованными с теми, что предоставлены существующим программам.
- Необходимо регулярно проверять атрибуты защиты файлов и каталогов командой **trustchk**.
- Минимальная длина паролей должны составлять 8 символов. За этим должен следить пользователь ISSO.
- У всех пользователей должна быть правильная начальная оболочка. За этим должен следить пользователь SA.
- ИД обычных пользователей должны отличаться от системных ИД. За этим должен следить пользователь SA. Системные ИД имеют uid меньше 128.

Конфигурация системы:

При настройке системы пользователи ISSO и SA должны выполнить некоторые действия. ISSO отвечает главным образом за защиту, а SA - за обычные задачи администрирования.

ISSO выполняет следующие задачи:

- Установка и настройка основных функций защиты, включая контроль, учет и защиту съемных устройств.
- Настройка стартовых сценариев `/etc/rc.mls` и `/etc/rc.mls.boot` для реализации стратегии защиты организации.

  Примечание: Все изменения, вносимые в стартовые сценарии системы, не являются частью апробированной конфигурации, и их необходимо проверить перед аккредитацией системы.
- Настройка учетных записей для входа в систему.
- Настройка паролей для входа в систему.
- Настройка диапазона SL для устройств tty, которые позволяют работать с SL, указанными для порта tty. Дополнительная информация приведена в описании команды **chsec**.
- Настройка SL системных устройств для магнитных лент и дискет. Дополнительная информация приведена в описании команды **setsecattr**.
- Настройка функций защиты системы в организации.

  Примечание: Все изменения, вносимые в настраиваемые функции защиты, не являются частью апробированной конфигурации, и их необходимо проверить перед аккредитацией системы. Изменение параметров конфигурации по умолчанию может привести к тому, что система будет работать в менее безопасном режиме.
- Настройка базы данных защиты для защищенной загрузки и защищенного восстановления. Дополнительная информация приведена в описании команды **trustchk**.
- Настройка групп в системе.

ISSO и SA совместно настраивают принтеры. SA настраивает принтеры в системе, а ISSO настраивает диапазон SL для принтеров.

Конфигурация сети:

ISSO отвечает прежде всего за защиту сети, а SA - за каждодневное администрирование сети. Для правильной настройки сети необходимы совместные действия ISSO и SA.

Во время установки системы Trusted AIX защите сети присваиваются параметры по умолчанию. Эта система может также передать метки секретности другим хостам Trusted AIX сети. ISSO устанавливает и настраивает базовые функции сети, предоставляемые вместе с системой. ISSO настраивает сетевые таблицы, а затем выполняет команду **tninit** для сохранения баз данных.

Сетевой доступ:

При подключении к системе, отличной от Trusted AIX, через сеть, либо к системе Trusted AIX, не использующей компонент Trusted Networking, некоторые атрибуты защиты не могут быть переданы системой, отличной от Trusted AIX. В этом случае система Trusted AIX применяет механизмы защиты по умолчанию. Эти механизмы устанавливаются системным администратором.

Настройка учетных записей пользователей:

ISSO и SA совместно настраивают учетные записи пользователей в системе. ISSO отвечает главным образом за атрибуты защиты пользователей, а SA - за обычные прочие атрибуты.

ISSO выполняет следующие задачи для каждого пользователя:
- Настройка допусков. Дополнительная информация приведена в описании команд **chsec** и **chuser**.
- Настройка ролей и прав доступа
- Настройка групп
- Настройка уровня допуска для домашнего каталога. Дополнительная информация приведена в описании команды **settxattr**
- Настройка паролей
- Настройка масок контроля

SA выполняет следующие задачи:
- Настройка учетных записей пользователей
- Информирование ISSO о новых учетных записях пользователей, для которых необходимо настроить атрибуты защиты

Конфигурация файловой системы:

Trusted AIX поддерживает большинство файловых систем, однако поддержка расширенных атрибутов защиты Trusted AIX в объектах файловой системы доступна только в JFS2 с EAv2.

Файловая система JFS2 с EAv1 преобразуется в EAv2 при монтировании в системе Trusted AIX. У файлов в этих файловых системах JFS2 нет атрибутов защиты. Для доступа к этим файлам система применяет атрибуты `SYSTEM_LOW` по умолчанию. Атрибуты защиты можно задавать в файлах командой **settxattr**.

В сетевой среде каталог в одной системе можно пометить как общий. В этом случае каталог можно будет смонтировать и обращаться к нему в других системах сети так, как если бы это был корневой каталог файловой системы в разделе локального диска.

Файловая система может быть многоуровневой (MLFS) или одноуровневой (SLFS). У каждого файлового объекта в MLFS свои метки, в то время как у всех объектов SLFS одни и те же метки, используемые в качестве точек монтирования. SLFS не поддерживает многоуровневые каталоги и разделенные каталоги.

Доступ к файловой системе:

Когда процесс пытается получить доступ к объекту файловой системы, система проверяет доступ к каждому компоненту пути.

Если у процесса нет прав на поиск ко всем каталогам пути, то процесс не сможет получить доступ к объекту. Когда используется относительный путь, доступ к текущему каталогу проверяется независимо от того, задана ли явная ссылка с использованием точки (`.`) на текущий каталог в начале имени пути.

Управление защищенной сетью:

Управление защищенной сетью включает в себя ряд аспектов, таких как настройка системы, настройка базы данных, синтаксис правил сети (netrule), создание правил, флаги защищенной сети и параметры RIPSO/CIPSO.

Предупреждение об изменении конфигурации по умолчанию:

Сетевые возможности AIX Trusted Network спроектированы таким образом, что допускают практически любую осмысленную конфигурацию. Тем не менее, изменение конфигурации по умолчанию без понимания того, как работает AIX Trusted Network, может представлять опасность.

Неправильная настройка компьютера может привести к автоматической частичной или полной утере или модификации информации о защите. По этой причине, не следует изменять значения по умолчанию в сетевых таблицах, не будучи знакомым с AIX Trusted Network.

База данных конфигурации защищенной сети AIX:

Конфигурация сети на момент загрузки определяется файлами `rules.host` и `rules.int`.

После стандартной установки Trusted AIX правила хоста и файлы правил отсутствуют. Для сохранения новых или обновленных правил в файлах можно воспользоваться командой **netrule** с флагом **-u**. Эти файлы являются двоичными базами данных, которыми можно управлять с помощью команды **tninit**. Для применения команды **tninit** пользователю необходимы права доступа `aix.mls.network.init`.

Просмотр базы данных правил AIX Trusted Network:

Содержимое набора базы данных правил AIX Trusted Network можно просмотреть с помощью действия **disp** команды **tninit**.

Введите следующую команду, чтобы добавить расширения **.host** и **.int** в *имя_файла* с целью сгенерировать имена файлов базы данных правил хостов и базы данных правил сетевых интерфейсов. Содержимое обоих файлов будет отправлено в стандартный поток вывода в удобочитаемой форме.

```
tninit disp имя_файла
```

Введите следующую команду для просмотра загрузочной конфигурации по умолчанию:

```
tninit disp /etc/security/rules
```

Загрузка базы данных правил AIX Trusted Network:

Команда **tninit** считывает набор баз данных правил AIX Trusted Network и загружает их в ядро, после чего этот набор становится активным. Имена файлов таблиц аккредитования хоста и интерфейса указываются в том же методе, что и действие **tninit disp**.

Необязательный флаг **-m** указывает, что система должна применять существующие правила хостов. Если флаг **-m** не указан, то все существующие правила хостов удаляются перед загрузкой нового активного

набора. Если флаг **-m** указан, то существующий и новый наборы правил хостов объединяются; при конфликте правил существующие правила заменяются на новые. Все правила интерфейсов заменяются независимо от того, указан ли флаг **-m**.

Следующая команда загружает новые правила, сохраняя старый набор правил:

```
tninit -m load /dir/dir/имя_файла
```

Эта команда добавляет расширения **.host** и **.int** к файлу *имя_файла*, создавая два файла, содержащие всю базу данных.

Сохранение базы данных защищенной сети AIX:

Для загрузки и сохранения базы данных правил используются похожие конструкции.

К указанному имени файла добавляются расширения **.int** и **.host**, и два создаваемых файла сохраняются в базе данных. Действие save команды **tninit** позволяет сохранить все активные правила, действующие в ядре.

Создать набор правил по умолчанию можно командой **netrule**. Эта команда настроит правила ядра согласно требуемой стратегии защиты среды, после чего можно будет вызвать команду **tninit**. Следующая команда создает файлы `/etc/security/rules.int` и `/etc/security/rules.host`:

```
tninit save /etc/security/rules
```

Конфигурация ядра защищенной сети AIX:

Обладая правами доступа `aix.mls.network.config`, с помощью команды **netrule** можно полностью настроить набор правил ядра защищенной сети AIX в соответствии со стратегией защиты сайта.

Командой **netrule** можно пользоваться для управления как правилами хоста, так и правилами сетевого интерфейса в ядре. Дополнительную информацию см. в описании команды **netrule**.

С каждым интерфейсом в системе должно быть связано определенное правило. Попытка удаления правила интерфейса ведет к возврату этого правила в состояние, заданное по умолчанию. При добавлении нового правила интерфейса, текущее правило будет перезаписано. Правило, заданное для интерфейса по умолчанию, можно просмотреть, запросив правило интерфейса с указанием имени интерфейса “default.” Например: `# netrule iq default`

Синтаксис netrule:

Для команды **netrule** существуют синтаксические правила хостов и интерфейсов.

В случае применения к хостам синтаксис команды **netrule** следующий:

netrule h l [**i** | **o** | **io**]

netrule h q { **i** | **o** } *спецификация_правила_хостов_src спецификация_правила_хостов_dst*

netrule h - [{ **i** | **o** } [**u**] [*спецификация_правила_хостов_src спецификация_правила_хостов_dst*]

netrule h + { **i** | **o** } [**u**] *спецификация_правила_хостов_src спецификация_правила_хостов_dst* [*флаги*] [*опции_RIPSO/CIPSO*] *защита*

В случае применения к интерфейсам синтаксис команды **netrule** следующий:

netrule i l

netrule i q *интерфейс*

netrule i + [**u**] *интерфейс* [*флаги*] [*опции_RIPSO/CIPSO*] *защита*

Первый элемент, `h` или `i`, указывает операцию хоста или сетевого интерфейса.

Следующим указано требуемое действие. Существует четыре возможных действия:

l Показать все правила

q Запросить конкретное правило

- Удалить правило хостов или вернуть правило интерфейсов в состояние по умолчанию

+ Добавить или переопределить правило

Третий элемент в правилах хостов идентифицирует тип правила. Для правил хостов существует различие между входящими и исходящими правилами. Входящие правила применяются ко всем входящим пакетам, исходящие - ко всем исходящим пакетам; `i` обозначает входящее правило, `o` - исходящее правило, а `io` или отсутствие опции, если применимо, - оба эти правила. Если при добавлении или удалении правила хостов или интерфейсов указан последний элемент `u`, то после успешного добавления или удаления правила хостов или интерфейсов файлы `/etc/security/rules.host` и `/etc/security/rules.int` обновляются.

Спецификация правил защищенной сети AIX:

Согласно правилам интерфейса, необходимо ввести имя интерфейса сети. Правила хоста являются гораздо более гибкими, и потому требуют более сложной спецификации правил.

Для указания интерфейса введите имя сетевого интерфейса, к которому следует применять данное правило. Именами сетевых интерфейсов являются имена типа `en0`. Просмотреть имена сетевых интерфейсов можно с помощью команды **ifconfig -a**. Конкретный интерфейс необходимо указывать только по имени. Для указания интерфейса не подходит имя порта, протокол или маска подсети.

Правила хоста требуют более сложной спецификации правил. Системой защищенной сети AIX используется наиболее конкретное из применяемых правил. Например, стратегия сайта может быть настроена таким образом, что правило хоста с маской 24 применяется ко всем хостам подсети, но к одному из хостов в сети применяется более конкретизированное правило, и этот хост будет использовать именно это специальное правило. В то же время, к отдельному порту TCP этого хоста может применяться другое более конкретное правило. Гибкость конфигурации защищенной сети AIX обеспечивает возможность реализации любой стратегии защиты сайта, необходимой для определенного приложения. Точный формат следующий:

исходный_хост [/*маска*] [**=** *proto*] [**:***начало_диапазона_портов* [**:***конец_диапазона_портов*]]

целевой_хост [/*маска*] [**=** *proto*] [**:***начало_диапазона_портов* [**:***конец_диапазона_портов*]]

исходный_хост
Имя исходного хоста, его адрес IPv4 или IPv6.

целевой_хост
Имя целевого хоста, его адрес IPv4 или IPv6.

маска Маска подсети. Номер указывает на количество значимых битов MSB. При записи адреса IPv4/пары подсети в формате *a.b.c.d/e*, *e* должно быть числом от 0 до 32. Это число указывает на количество единиц в начале маски подсети. Например, для адреса IPv4 `/24` означает маску сети `255.255.255.0`, которая в 32-разрядном формате имеет вид `11111111.11111111.11111111.00000000` - 24 единицы и восемь нулей.

proto Номер или название протокола, записанное в файле `/etc/protocols` (например, `=tcp`).

начало_диапазона_портов
Порт TCP или UDP, к которому применяется данное правило, или начало диапазона портов, если правило применяется к нескольким портам. Это может быть либо номер порта, либо имя службы UDP или TCP, согласно записи в файле `/etc/services`.

конец_диапазона_портов
Верхний предел диапазона портов.

Описание флагов защищенной сети AIX:

В системе защищенной сети AIX имеется два кластера флагов. Если они не указаны, используются значения, предусмотренные по умолчанию.

Флаги **-d** и **-r** употребляются следующим образом:

-d *drop*

drop Защищенную сеть AIX можно настроить для отбрасывания всех пакетов

r Отбрасывать все пакеты в этом интерфейсе

n Запрещается автоматически отбрасывать все пакеты в этом интерфейсе (параметр интерфейса по умолчанию)

i Использовать параметр интерфейса по умолчанию (хост по умолчанию, только хост)

-f*rflag***:***tflag*

rflag Требование опции защиты к поступающим (полученным) пакетам

r только RIPSO

c только CIPSO

e CIPSO или RIPSO

n Ни CIPSO,ни RIPSO (параметр системы по умолчанию)

a Ограничения отсутствуют

i Использовать параметр интерфейса/системы по умолчанию (значение по умолчанию)

tflag Обработка опции защиты исходящих (переданных) пакетов

r Ограничение RIPSO на все IP-заголовки исходящего пакета

c Ограничение CIPSO на все IP-заголовки исходящего пакета

i Использовать параметр интерфейса по умолчанию (хост по умолчанию, только хост)

Опции RIPSO/CIPSO:

Подсистема AIX Trusted Network поддерживает опции настройки меток пакетов CIPSO и RIPSO.

-rpafs=*поле_PAF* [**,** *поле_PAF* ...]

Указывает каждое *поле_PAF*, принимаемое при получении пакетов IPSO. Таких полей может быть до 256.

-epaf=*поле_PAF*

Указывает *поле_PAF*, подключаемое к ошибочным ответам, когда ошибочные пакеты отправляются с использованием IPSO переданных пакетов.

-tpaf=*поле_PAF*

Указывает *поле_PAF*, применяемое к исходящим пакетам, когда IPSO используется в переданных пакетах.

*Поле_PAF***:NONE** | *PAF* [**+** *PAF* ...]

Поле_PAF - это совокупность *PAF*. Существует пять отдельных *PAF*, которые можно объединить в *поле_PAF*. Это **GENSER**, **SIOP-ESI**, **SCI**, **NSA** и **DOE**. *Поле_PAF* - это сочетание этих значений, разделенных знаком плюс (**+**). Например, *поле_PAF*, содержащее **GENSER** и **SCI**, представляется как **GENSER+SCI**. Допустимо специальное *поле_PAF* **NONE**; это означает *поле_PAF* без *PAF*.

-DOI=*doi*

Задает домен интерпретации для пакетов CIPSO. У входящих пакетов CIPSO должен быть этот **DOI**, а исходящие пакеты CIPSO будут помечены этим **DOI**.

-tags=*тег*[**,***тег* ...]

*тег***=1 | 2 | 5**

Задает набор тегов, принимаемых и доступных для передачи опциями CIPSO. Это сочетание цифр **1**, **2** и **5**, разделенных запятыми. Например, **1,2** включает теги **1** и **2**.

Стратегия защиты защищенной сети AIX:

Необходимо указать минимальную, максимальную и заданную по умолчанию метку чувствительности (SL).

Подразумеваемая или заданная по умолчанию метка чувствительности применяется ко всем пакетам, не включающим информацию о собственных метках. Формат указания уровней меток следующий:

+*min* **+***max* **+***default*

Допускается использование любой метки, разрешенной согласно файлу кодировки меток. Квоты не обязательны для меток, включающих в себя пробелы.

Примеры netrule:

Ниже приведены примеры команды **netrule**.

Следующая команда настраивает **en0** так, чтобы он не передавал опции защиты и пропускал все пакеты:

```
netrule i+ en0 +impl_lo +ts all +impl_lo
```

Следующая команда настраивает хост **185.0.0.62** так, чтобы он принимал только пакеты CIPSO в диапазоне между **CONFIDENTIAL A** и **TOP SECRET ALL**:

```
netrule h+i 192.168.0.0 /24 185.0.0.62 -fc:c +confidential a +top secret all +confidential a
```

Следующая команда удаляет все пакеты telnet из подсети:

```
netrule h+i 192.168.0.0 /24 =tcp :telnet 192.0.0.5 -dr +impl_lo +impl_lo +impl_lo
```

Дополнительная информация и примеры приведены в описании команды **netrule**.

Управление учетными записями пользователей:

Информация об идентификации и проверке подлинности (I&A) для каждого пользователя является защищенной и используется для уникальной идентификации пользователей и предоставления им прав доступа в системе.

Информация о пользователе включает в себя имя пользователя, текстовое имя ИД входа в систему, ИД пользователя, ИД группы, домашний каталог, пароль, данные о сроке действия пароля, оболочку, допуска, права доступа и маску контроля. Большинство этих данных сохраняются в следующих файлах:

/etc/passwd
Имена пользователя, ИД, основные группы и домашние каталоги

/etc/group
Дополнительные группы и домашние каталоги

/etc/security/passwd
Пароли пользователей в зашифрованном виде

/etc/security/user
Ограничения входа в систему, параметры пароля (например, минимальная длина), umask и т.д.

Файлы /etc/security/passwd и /etc/security/user недоступны для чтения обычными пользователями. Для защиты файла /etc/security/passwd включены бит неизбирательного доступа и SL SYSTEM_HIGH. Запрет чтения файла зашифрованных паролей предотвращает атаки с подбором пароля методом перебора.

Пользователи с соответствующими правами доступа могут изменять эти файлы и работать напрямую с этими файлами, но обычно удобнее делать это с помощью команды **smit**. Команда **smit** вызывает программу SMIT, в меню которой можно выбрать задачи управления системой, например, управления пользователями.

ИД пользователей и групп:

Есть два класса ИД пользователей: системные ИД и ИД обычных пользователей. Системные ИД резервируются для владельцев защищенных подсистем и специальных функций администрирования системы. Обычные ИД присваиваются пользователям для интерактивной работы в системе.

Каждый пользователь имеет свой уникальный ИД в системе. Пользователи могут быть присвоены также ИД группы, используемые всеми пользователями группы и не являющиеся уникальными. Диапазон значений ИД ограничен. В следующей таблицы описаны возможные значения ИД. Эти значения достаточны для всех системных, обычных и групповых ИД.

Системные ИД
0 - 127

Обычные ИД
128 - MAXUID

Групповые ИД
0 - MAXUID-1

Значение MAXUID определено в файле /usr/include/sys/param.h

Присваивать ИД новым пользователям следует с осторожностью. Если обычному пользователю будет присвоен ИД со значением меньше 128, то пользователь не сможет войти в систему.

ИД пользователей не следует использовать повторно. При удалении пользователя рекомендуется оставить его запись в файлах /etc/passwd и /etc/security/passwd и заблокировать эту учетную запись. Это можно сделать с помощью команды **smit**. При этом пользователь не сможет войти в систему, а ИД не будет использоваться повторно. Это предотвращает доступ нового пользователя к старым файлам прежнего пользователя, если они еще не были удалены. Также не будет возникать никакой неоднозначности в процедурах контроля.

Для работы с файлами /etc/passwd, /etc/security/passwd и /etc/group применяются команды **mkuser**, **chuser**, **rmuser**, **pwdadm** и **passwd**. Эти команды учитывают все указанные выше рекомендации и прочие соображения, связанные с защитой системы. Команда **mkuser** позволяет добавить в систему только обычных пользователей.

Примечание: Рекомендуется придерживаться следующих правил:

- Никогда не присваивайте ИД старого пользователя новому пользователю
- Никогда не присваивайте пользователям одинаковые ИД
- Никогда не присваивайте системный ИД обычному пользователю
- Никогда не присваивайте MAXUID пользователю или группе

Пароль:

Пароль - это строка символов, связанная с пользователем и применяемая для проверки его подлинности при запуске сеанса.

Пароль хранится в зашифрованном виде в теневом файле. Незашифрованный пароль в системе не встречается.

Примечание: Пароли ролевых пользователей играют исключительно важную роль в защите системы и должны быть надежно защищены в любое время.

Требования к возрасту паролей:

Пользователи могут изменять свои пароли, при условии соблюдения требований к возрасту паролей.

Требования к возрасту паролей обязывают пользователя сменить свой пароль, если он просуществовал в системе определенное время. Требования к возрасту паролей включают минимальный возраст и максимальный возраст. Пароль нельзя изменить до достижения им минимального возраста. Пароль необходимо изменить после достижения им максимального возраста.

Параметры возраста паролей задаются в файле `/etc/security/user`. С требованиями к возрасту паролей связаны следующие параметры:

maxage
Максимальная продолжительность действия пароля в неделях

maxexpired
Максимальное время в неделях сверх maxage, в течение которого пароль может быть изменен пользователем

minage
Минимальное время в неделях между сменами пароля

minlen Минимальная длина пароля

Можно задать и другие параметры, определяющие допустимые символы в пароле. Полный список параметров пароля приведен в описании команды **passwd**.

Оболочка:

Обычно при работе с приложением, таким как текстовый процессор или электронная таблица, нет необходимости обращаться напрямую к функциям операционной системы, так как за это отвечает само приложение. Однако в некоторых случаях требуется обращаться напрямую к функциям операционной системы, в обход интерфейса приложения.

Для прямого взаимодействия с операционной системой используется программа оболочки. В программе оболочки пользователи могут вводить команды AIX, обращаться прямо к файлам и каталогам и выполнять другие операции. Оболочка по умолчанию указывается для пользователя в файле `/etc/passwd`. Эта оболочка по умолчанию (`/bin/sh`, `/bin/csh` или `/bin/ksh`) вызывается программой **login** или **xterm**, когда пользователю требуются функции оболочки.

Действующие SL и TL входа в систему:

Пользователям присваиваются SL и TL входа в систему по умолчанию. SL и TL входа в систему по умолчанию - это действующая SL и действующая TL процесса пользователя после успешного входа в систему.

Если пользователь не хочет входить в систему с SL входа в систему по умолчанию, то он может выбрать другую SL во время входа в систему с помощью опции **-e** команды **login**. SL, указываемая пользователем, должна поглощаться допуском пользователя и входить в диапазон аккредитования системы. Пользователь может указать TL во время входа в систему с помощью опции **-t** команды **login**.

SL и TL входа в систему по умолчанию задаются в файле `/etc/security/user` вместе с именем пользователя и допуском для каждого пользователя. Действующая SL пользователя должна лежать в диапазоне SL терминала, заданном в файле `/etc/security/login.cfg`. Действующая SL пользователя должна поглощаться максимальной SL терминала и поглощать минимальную SL. Действующая TL пользователя должна совпадать с TL терминала.

Допуски:

Во время входа пользователя в систему оболочке его процесса назначаются шесть меток.

Действующая SL применяется системой в проверках MAC. Минимальный и максимальный допуски SL ограничивают действующую SL; действующая SL не может переопределять максимальный допуск SL, но должен переопределять минимальный допуск SL. Действующая TL применяется системой в проверках MIC. Минимальный и максимальный допуски TL ограничивают действующую TL; действующая TL не может переопределять максимальный допуск TL, но должен переопределять минимальный допуск TL.

Пользователь с правами ISSO может изменять допуск SL, допуск TL, SL входа в систему по умолчанию и TL входа в систему по умолчанию для любого пользователя. Все эти значения задаются в файле `/etc/security/user`.

Разделение ответственности за информацию о пользователе:

Отдельный пользователь не может добавить пользователя в систему. Пользователи добавляются в систему в результате совместных действий пользователей с правами SA и ISSO.

Пользователь с правами SA может добавить несекретную информацию о пользователе, включающую имя пользователя, ИД пользователя, ИД группы, текстовое имя ИД входа в систему, оболочку и домашний каталог. Пользователь с правами ISSO может добавить секретную информацию о пользователе, включающую пароль пользователя, допуск, маску контроля и роли. Требование об участии двух человек в процедуре добавления пользователя не позволяет отдельному пользователю, имеющему широкие права доступа, предоставлять общесистемные права доступа другому пользователю.

Расширенный контроль:

Trusted AIX расширил функции работы с защитой в подсистеме контроля.

Новые поля записей контроля:

Следующие поля были добавлены во все записи контроля AIX для Trusted AIX. Эти новые поля могут применяться в команде **auditselect** в качестве критериев выбора.

- Роли контролируемого процесса
- Действующий TL контролируемого процесса или объекта
- Действующий SL контролируемого процесса или объекта
- Эффективные привилегии контролируемого процесса

Trusted AIX также контролирует следующие атрибуты защиты в некоторых журналах контроля:

- TL контролируемого процесса или объекта
- SL контролируемого процесса или объекта
- Флаги защиты, относящиеся к Trusted AIX

Эти новые атрибуты защиты можно просмотреть командой **auditpr -v**.

Контрольные диапазоны:

Trusted AIX предусматривает механизм, позволяющий администраторам задавать набор контрольных диапазонов на основе TL и/или SL контролируемых процессов или объектов. Все объекты и субъекты, TL или SL которых выходят за границы контрольных диапазонов, будут игнорироваться.

Для того чтобы задать контрольные диапазоны для процессов и объектов, добавьте раздел **war** в файл `/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
```

obj_min_sl и **obj_max_sl** определяют контрольный диапазон SL для объектов. **sub_min_sl** и **sub_max_sl** - контрольный диапазон SL для субъектов (процессов). **obj_min_tl** и **obj_max_tl** - контрольный диапазон TL для объектов. **sub_min_tl** и **sub_max_tl** - контрольный диапазон TL для субъектов (процессов).

Раздел **war** считывается командой **audit start** и загружается в ядро перед запуском подсистемы контроля. Если раздел **war** опущен, текущие контрольные диапазоны в ядре удаляются. Ядро не выполняет никаких проверок контрольных диапазонов TL или SL, если в нем отсутствуют контрольные диапазоны TL и SL.

Флаг ядра Trusted AIX:

Если при установке система настроена как система Trusted AIX, в переменной **_system_configuration** активируется флаг глобального ядра. В ядре имеется макрокоманда **__MLS_KERNEL()**, позволяющая определить, настроена ли система как Trusted AIX. Макрокоманду вызывают прикладные программы пользовательской области или процедуры ядра. Если макрокоманда **__MLS_KERNEL()** возвращает значение **1**, значит, система настроена как Trusted AIX. Все остальные возвращаемые значения указывают на то, что система не настроена как Trusted AIX.

Обновление существующих программ:

Существующие приоритетные или защищенные программы, как правило, корректно работают в защищенной системе, не требуя изменений.

Однако в целях улучшения защиты и/или совместимости этих программ с новыми версиями могут вноситься определенные изменения. Многие рекомендации по созданию новых программ также касаются обновления существующих программ. В частности, имеются такие рекомендации:

- Программы, проверяющие привилегированность процессов (то есть, равен ли ИД эффективного пользователя 0), необходимо модифицировать в соответствии с рекомендациями из раздела Проверка прямых привилегий
- Код, обрабатывающий разряды полномочий (разряды режима) в стандартной системе UNIX, должен изменяться в зависимости от наличия списков ACL

- Код, который обычно выполнялся с заданием_uid_как_root, необходимо проверить на использование прав доступа, и ему необходимо присвоить соответствующие права доступа

Резервное копирование и восстановление:

При импорте и экспорте данных в Trusted AIX системах используются защищенные версии команд **backup** и **restore**.

Команды **backup** and **restore** были расширены для работы с метками. Эти расширения прозрачны для пользователя и, не считая расширений для работы с метками, команды работают так же, как и стандартные команды **backup** и **restore** системы AIX. Для того чтобы отключить резервное копирование или восстановление расширенной информации защиты, можно использовать флаг **-O**.

Система импорта/экспорта защищена за счет соединения механизмов привилегий и идентификации.

Ограничения команды cron:

Команда **cron** отключена и не будет запускать никаких заданий, когда система находится в режиме настройки. Если система находится в рабочем режиме, команда **cron** запускает задания с той меткой секретности, с которой задание было передано на выполнение, и пользовательской меткой целостности по умолчанию.

Существуют ограничения, например минимальный и максимальный допуски пользователя. В зависимости от того, какое из этих значений задано последним, допуск берется либо из параметров времени передачи задания на выполнения, либо времени последнего перезапуска команды **cron**. Администрировать команду **cron** может только пользователь с правами SA.

Монтирование файловых систем и доступ к ним:

Trusted AIX поддерживает метки (SL и TL) в файловых системах JFS2 с EAv2. Пользователь с правами SA или SO при необходимости может смонтировать файловую систему, не поддерживающую метки (CDFS или HSFS). В этом случае у файлов в смонтированной файловой системе не будет отдельных SL, TL или FSF; вместо этого, файлы унаследуют атрибуты защиты точки монтирования.

Управление системой Trusted AIX

Для обеспечения защиты системы Trusted AIX необходимо соблюдать рекомендации по правильному управлению системой.

Управление системой Trusted AIX выполняется определенными пользователями, учетные записи которых связаны с административными ролями. Эти пользователи называются Information System Security Officer (ISSO), System Administrator (SA) и System Officer (SO), и у каждого из них есть права доступа, позволяющие им выполнять конкретное подмножество административных задач. Эти пользователи связаны с системными ролями `isso`, `sa` и `so`, соответственно. Термины ISSO, SA и SO применяются для обозначения пользователей с ролями `isso`, `sa` и `so`, соответственно. Некоторые административные задачи могут быть выполнены только в результате совместных действий двух из этих трех пользователей-администраторов, поскольку у одного администратора нет достаточных прав для выполнения этих задач. Например, при добавлении нового пользователя в систему только пользователь SA может добавить новую учетную запись пользователя и только пользователь ISSO может задать пароль, допуск и контрольную маску пользователя. Это разделение обязанностей называется правилом двух человек.

Примечание: Эффективность правила двух человек зависит от прав доступа, присвоенных административным ролям. Присвоение административным ролям больших прав доступа, чем это необходимо, может сделать систему уязвимой к атакам инсайдеров. Дополнительная информация о присвоении прав доступа ролям приведена в разделе RBAC.

Системные роли `isso`, `sa` и `so` по умолчанию связаны со следующими правами доступа Trusted AIX. При изменении этих связей необходимо принять соответствующие меры, поскольку система может стать уязвимой.

Таблица 41. Роли и права доступа

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		

Управление системой пользователями с правами Information System Security Officer:

Система Trusted AIX управляется согласованными действиями пользователей с правами ISSO, SA и SO.

Во время установки Trusted AIX создаются три учетные записи пользователей **isso**, **sa** и **so** (если только они уже не существуют в результате перехода от обычной версии AIX к Trusted AIX). Эти пользователи связаны с ролями isso, sa и so, соответственно.

Примечание: Учетные записи по умолчанию предназначены только для начальной установки и настройки системы Trusted AIX. Рекомендуется присвоить эти роли обычным пользователям. После присвоения этих ролей другим пользователям учетные записи по умолчанию можно удалить. Дополнительная информация об установке Trusted AIX приведена в разделе *Установка и миграция*.

Операции ISSO

Основная обязанность пользователя с правами Information System Security Officer (ISSO) - управление защитой системы. Выполнять операции ISSO может только пользователь с правами ISSO. Это следующие операции:

- Планирование, реализация и внедрение стратегии защиты сайта
- Установка общесистемных значений по умолчанию для допусков, прав доступа, привилегий, управляющих элементов входа в систему и паролей пользователей
- Настройка профайлов идентификации пользователей, отражающих уровень доверия к пользователям, который учитывается при создании их учетных записей системным администратором
- Присвоение атрибутов защиты, SL и TL устройствам, таким как терминалы, принтеры, съемные дисководы и накопители магнитных лент
- Присвоение флагов защиты, меток, привилегий и наборов прав доступа файлам
- Восстановление системы в надежное состояние в случае ее сбоя

Управление системой контроля:

Доступ к командам контроля ограничен и предоставляется только пользователям с правами доступа **AUDITSYS**. Дополнительная информация приведена в описании команд **audit**, **auditselect** и **auditpr**.

В следующем примере описан порядок:

1. Создания файловой системы, которая будет использована для файлов контрольного журнала
2. Запуска системы контроля
3. Порождения некоторых записей
4. Синтаксического анализа контрольного журнала для получения разных типов записей

Выполните следующие команды - для этого необходимо иметь права доступа **FSADMIN**:

```
/usr/sbin/crfs -v jfs -g rootvg -m /audit -a size=32M -A yes
mount /audit
```

С помощью команды **/tbin/auctlmod -e** добавьте следующую запись в пользовательский раздел файла `/etc/security/audit/config`:

```
username = ALL
```

Замените *username* именем реального пользователя, который может войти в систему.

Создайте файл (это может сделать пользователь ISSO) с именем `/tmp/top_secret` и замените SL файла на **TS ALL**.

```
touch /tmp/top_secret
/usr/sbin/settxattr -f sl= "TS ALL" /tmp/top_secret
```

Выполните следующую команду - для этого необходимо иметь права доступа **AUDITSYS**:

```
/usr/sbin/audit start
```

Контрольная система установлена и запущена; теперь она будет регистрировать действия пользователя *username*, как только он войдет в систему.

Войдите в систему как пользователь под именем *username*, указанным в файле `/etc/security/audit/config`, и выполните следующие команды:

```
ls -l /tmp/top_secret
exit
```

Выполните следующие команды - для этого необходимо иметь права доступа **AUDITSYS**:

```
audit shutdown
$ /usr/sbin/auditselect -e "mac_fail==WILDCARD" /audit/trail | \
/usr/sbin/auditpr -v -APSV > /tmp/audit_trail-mac_failure
```

Просмотрите контрольный журнал, который был перенаправлен в файл `/tmp/audit_trail-mac_failure`, и найдите **mac_fai**l. Функция auditselect изменяется, принимая следующие параметры:

- **subj_sl**
- **obj_sl**
- **mac_fail**
- **mac_pass**
- **mic_fail**
- **mic_pass**
- **priv_fail**
- **priv_pass**

- **auth_pass**
- **fsf_fail**
- **fsf_pass**

Во всех указанных параметрах используется слово **WILDCARD** в качестве согласованного значения.

Управление метками объектов и процессов:

С каждым объектом файловой системы и системным процессом связаны метки.

У всех объектов файловой системы, отличных от обычных файлов, есть диапазон меток секретности и метка целостности. У процессов есть и диапазон меток секретности, и диапазон меток целостности. Помимо этих диапазонов, у процессов есть действующая SL и действующая TL. Эта метка указывает текущую SL или TL, с которой работает процесс. Просмотреть метки можно командой **lstxattr**. Задать метки объектов файловой системы и процессов можно командой **settxattr**.

Управление защитой сети:

Для AIX Trusted Network необходимо, чтобы ISSO определил несколько таблиц. Эти таблицы хранятся в каталоге /etc/security. С помощью команды **tninit** создается двоичная версия, которая затем загружается в ядро.

Правила хостов и сетевых интерфейсов определяют, каким образом система обрабатывает входящие и исходящие сетевые пакеты. Правила хостов применяются к конкретным хостам. Правила сетевых интерфейсов - к интерфейсам, через которые хосты подсоединены к сети. В случае конфликта между правилом хостов и правилом интерфейсов приоритет отдается правилу хостов.

Для добавления, изменения и запроса правил служит команда **netrule**. В общем случае, правила определяют используемые протоколы, диапазоны адресов (хостов и портов), к которым применяются правила, и SL, которые присваиваются пакетам. Дополнительная информация приведена в описании команды **netrule**.

Команда **tninit** позволяет инициализировать подсистему AIX Trusted Network, сохранить правила в двоичном формате и просмотреть правила в текстовом формате.

Настраиваемые функции защиты:

Параметры настраиваемых функций показываются в ходе загрузки.

Настраиваемые параметры сохраняются в ODM. Эти параметры можно просмотреть командой **getsecconf** и изменить командой **setsecconf**, выполняемой от имени пользователя ISSO.

Управление метками:

Пользователь с правами ISSO может добавлять, изменять и удалять закодированные метки, редактируя файл `/etc/security/enc/LabelEncodings`. Файл `/etc/security/enc/LabelEncodings` определяет, каким образом обычные имена преобразуются в двоичное представление системных меток секретности.

Примечание: Изменение файла закодированных меток секретности в работающей системе может вызвать появление недопустимых меток, если только не предприняты исключительные меры предосторожности. Поскольку объекты могут быть помечены отдельными словами или сочетаниями слов, подчиняющимися ограничениям, необдуманное изменение, добавление или удаление ограничений на сочетания слов может вызвать появление недопустимых меток.

Файл `/etc/security/enc/LabelEncodings` преобразуется в двоичную форму библиотечной процедурой **l_init** и хранится в таблицах. Эти таблицы служат для преобразования SL, баннеров принтеров и допусков во внутреннюю двоичную кодировку и обратно.

Реализация меток в Trusted AIX основана на использовании программы MITRE Compartmented Mode Workstation Labeling. В документе Compartmented Mode Workstation Labeling: Encodings Format, DDS-2600-6216-93 (MTR 10649 revision 1) от сентября 1993 г. разъясняется стандартный формат кодирования меток.

В стандартном формате кодирования меток метки целостности и секретности рассматриваются так же, как в разделе **Sensitivity Labels** файла `/etc/security/enc/LabelEncodings`.

Trusted AIX поддерживает (необязательно) раздел **Integrity Labels**, позволяющий меткам целостности отличаться от меток секретности.

Управление разделенными каталогами:

Для обычного пользовательского процесса разделенный каталог выглядит и функционирует так же, как обычный каталог. Однако в случае разделенного каталога разные процессы с разными SL видят разное содержимое одного и того же каталога.

Например, если процесс, работающий с меткой секретности **SECRET**, создает файл с именем **foo** в разделенном каталоге, то второй процесс, работающий с меткой секретности TOP SECRET, не увидит файл **foo** в этом каталоге и не сможет работать с ним. Кроме того, второй процесс может создать собственный файл **foo**, не конфликтующий с первым файлом **foo**.

Это достигается за счет использования скрытых подкаталогов. Для каждой уникальной SL, с которой процесс обращается к разделенному каталогу, существует разделенный подкаталог. Когда процесс обращается к разделенному каталогу, система автоматически перенаправляет его на скрытый подкаталог. В приведенном выше примере два файла **foo** на самом деле находятся в разных подкаталогах, хотя пользователю и кажется, что они находятся в одном каталоге.

Дополнительная информация о разделенных каталогах приведена в разделе “Разделенные каталоги” на стр. 429.

Разделенные каталоги поддерживаются в JFS2 с EAv2.

Создание разделенного каталога:

При создании разделенного каталога диапазон SL по умолчанию - от SYSTEM LOW SL до SYSTEM HIGH SL. Когда происходит обращение к разделенному каталогу, ядро автоматически создает дочерний каталог по метке (если он не существует) и перенаправляет на него пользовательский процесс.

Команда **pdmkdir** создает разделенный каталог. Команде **pdmkdir** необходимы права **aix.mls.pdir.create** для переопределения ограничений DAC, MAC и MIC. Команда **pdrmdir** удаляет пустой разделенный каталог.

Разделенные подкаталоги и подподкаталоги

Разделенные подкаталоги - это дочерние каталоги с метками для разделенного каталога. Когда процесс создает дочерний каталог в разделенном подкаталоге (с помощью команды **mkdir**), этот дочерний каталог становится разделенным подподкаталогом.

Создаваемый разделенный подкаталог наследует атрибуты защиты своего родительского разделенного каталога, за исключением минимальной SL и максимальной SL. В качестве минимальной SL и максимальной SL берется действующая SL первого процесса виртуального режима, обращающегося к разделенному каталогу.

Trusted AIX распознает четыре различных типа каталогов:

- обычный каталог (dir)

- разделенный каталог (pdir)
- разделенный подкаталог (psdir)
- разделенный подподкаталог (pssdir)

Виртуальный режим и реальный режим:

Существует два режима доступа к разделенным каталогам: виртуальный режим и реальный режим.

В виртуальном режиме процесс, обращающийся к разделенному каталогу, видит только содержимое разделенного подкаталога согласно своей метке. Процесс не видит разделенный каталог при работе в виртуальном режиме. Процесс видит разделенный каталог при работе в реальном режиме. В реальном режиме процессам доступно все содержимое разделенных каталогов и разделенных подкаталогов. Для процессов реального режима система не выполняет никаких перенаправлений.

По умолчанию процессы работают в виртуальном режиме. Реальный режим предназначен только для задач администрирования файловой системы. Команда **pdmode** позволяет выполнить процессы в режиме, отличном от текущего режима оболочки, или перевести оболочку в другой режим.

Хотя в реальном режиме пользовательский процесс может работать с разделенными каталогами и подкаталогами, этот тип доступа следует применять с осторожностью. Например, если в разделенный каталог помещается обычный каталог процессом реального режима, то этот каталог вообще не будет доступен процессам в виртуальном режиме.

Несмотря на то, что разделенный каталог выглядит как обычный каталог для процесса виртуального режима, для него существуют определенные ограничения.

Иерархия:

Существует иерархия разделенных каталогов и подкаталогов.

Иерархия разделенных каталогов и подкаталогов подчиняется следующим правилам:
- Каталог должен принадлежать одному из следующих четырех типов:
  - обычный каталог
  - разделенный каталог
  - разделенный подкаталог
  - разделенный подподкаталог
- Каталог может принадлежать только одному из этих типов в каждый момент времени
- Родительский каталог разделенного подкаталога должен быть разделенным каталогом
- Каждый дочерний каталог разделенного подкаталога должен быть разделенным подподкаталогом
- Родительский каталог разделенного подподкаталога должен быть разделенным подкаталогом

Любое нарушение этих правил приводит к тому, что дерево разделенных каталогов становится недопустимым, а файловая система - несогласованной, с непредсказуемым поведением.

Монтирование файловых систем:

Разделенный каталог или подкаталог может быть точкой монтирования, но разделенный подподкаталог - нет. Аналогично, корневой каталог монтируемой файловой системы может быть разделенным каталогом или подкаталогом, но не может быть разделенным подподкаталогом.

Создание и удаление каталогов:

Если процесс виртуального режима выполняется в разделенном подподкаталоге, команда **mkdir** создает обычный каталог. Если тот же процесс находится в разделенном подкаталоге и выполняет команду **mkdir**, то автоматически создается разделенный подподкаталог. Любой пустой каталог можно удалить, при условии соблюдения ограничений MAC, MIC и DAC.

Перемещение каталогов:

При перемещении каталогов действуют ограничения MAC, MIC и DAC.

Обычный каталог можно переместить в любое место. Если его новый родительский каталог - это разделенный подкаталог, то перемещенный обычный каталог станет разделенным подподкаталогом. В противном случае он останется обычным каталогом. Если его новый родительский каталог - это разделенный каталог и его имя конфликтует с именем потенциального разделенного подкаталога, то любая последующая попытка перенаправить процесс виртуального режима на этот потенциальный разделенный подкаталог потерпит неудачу.

Разделенный каталог можно переместить в другой обычный каталог, и после перемещения он останется разделенным каталогом. Вложенные разделенные каталоги не поддерживаются в Trusted AIX, поскольку они не дают никаких дополнительных преимуществ.

Разделенный подкаталог можно переместить только в разделенный каталог, и после перемещения он останется разделенным подкаталогом. Перемещение разделенного подкаталога в обычный каталог, разделенный подкаталог или разделенный подподкаталог запрещено.

Разделенный подподкаталог можно переместить в любое место. Если его новый родительский каталог - это обычный каталог, разделенный каталог или разделенный подподкаталог, то он станет обычным каталогом. В противном случае он останется разделенным подподкаталогом.

Таблица 42. Обзор перемещения каталогов

Перемещение каталога типа	**В обычный каталог**	**В разделенный каталог**	**В разделенный подкаталог**	**В разделенный подподкаталог**
Обычный	Допустимо. Остается обычным каталогом.	Допустимо[1]. Остается обычным каталогом.	Допустимо[1]. Становится разделенным подподкаталогом.	Допустимо. Остается обычным каталогом.
Разделенный	Допустимо. Остается обычным каталогом.	Допустимо[1]. Остается обычным каталогом.	Недопустимо.	Допустимо. Остается обычным каталогом.
Разделенный подкаталог	Недопустимо.	Допустимо. Остается разделенным подкаталогом.	Недопустимо.	Недопустимо.
Разделенный подподкаталог	Допустимо. Становится обычным каталогом.	Допустимо. Становится обычным каталогом.	Допустимо. Остается подподкаталогом.	Допустимо. Становится обычным каталогом.

[1] Если имя конфликтует с именем потенциального (не существующего в настоящее время) разделенного подкаталога, то любая последующая попытка перенаправить процесс виртуального режима на этот разделенный подкаталог потерпит неудачу.

Изменение типа каталога:

Команда **pdset** изменяет обычный каталог на разделенный. Команда, изменяющая разделенный каталог на обычный, не предусмотрена.

Замена номеров inode:

При обращении к разделенному подкаталогу, когда требуется его номер inode или номер inode его родительского разделенного каталога (..), возвращается номер inode его родительского разделенного каталога или номер inode родительского каталога его родительского разделенного каталога, соответственно. При обращении к разделенному подподкаталогу, когда требуется номер inode родительского каталога разделенного подподкаталога (..), возвращается номер inode родительского каталога его родительского каталога разделенного каталога.

Команды разделенных каталогов:

Эти команды применяются к разделенным каталогам.

pdmkdir
Создать разделенные каталоги

pdrmdir
Удалить разделенные каталоги и подкаталоги

pdlink Связать файлы в разделенных подкаталогах

pdset Задать каталоги как разделенные

pdmode
Восстановить текущий режим доступа к каталогу

Выполнить команду с указанным режимом доступа к каталогу

Обычный каталог, преобразованный в разделенный, можно вновь превратить в обычный.

Защита системы - обзор:

В обязанности ISSO входит мониторинг состояния защиты системы. Обзор защиты системы необходимо выполнить сразу же после установки и в любое другое время, когда целостность системы может оказаться под вопросом. Затем обзор защиты системы нужно выполнять регулярно.

База данных целостности системы хранится в файле `/etc/security/tsd/tsd.dat` и содержит информацию для защиты объектов файловой системы, таких как критические команды и системные устройства. Эту базу данных необходимо обновлять при добавлении нового устройства или при изменении данных о защите файлов. Дополнительная информация приведена в описании команды **trustchk**.

Команда **trustchk** сравнивает текущие параметры защиты файла, каталога или устройства с соответствующими данными в базе данных целостности системы, и исправляет все отклонения атрибутов защиты. Команда **trustchk** должна выполняться пользователем с правами ISSO.

Управление TTY:

Минимальные SL, максимальные SL и TL для устройств tty определяются в базе данных tty в файле `/etc/login`.cfg. Дополнительная информация приведена в описании команды **chsec**.

Эффективный SL пользователя, входящего в систему через порт TTY, должен быть в пределах диапазона, указанного для этого порта в этом файле. Если для порта TTY указана TL, отличная от NOTL, то эффективная TL пользователя должна совпадать с этой TL.

Управление допусками пользователей:

Для входа в систему все пользователи должны иметь метки, в том числе пользователи ISSO, SA и SO. Допуски пользователей можно указать в файле `/etc/security/user` как часть раздела пользователя. Атрибуты **minsl**, **maxsl**, **defsl**, **mintl**, **maxtl** и **deftl** задают минимальную SL, максимальную SL, SL по

умолчанию, минимальную TL, максимальную TL и TL по умолчанию соответственно. Если эти атрибуты указаны в разделе пользователя, то значения, указанные в разделе по умолчанию файла, присваиваются пользователю.

Изменять базу данных допусков может только пользователь ISSO. Просмотреть допуск пользователя можно командами **lsuser** и **lssec**, а изменить - командами **chuser** и **chsec**.

Значение SL по умолчанию должно поглощаться максимальным значением SL и поглощать минимальное значение SL. Аналогично, значение TL по умолчанию должно поглощаться максимальным значением TL и поглощать минимальное значение TL.

Примечание: Для того чтобы пользователь мог успешно войти в систему, должны выполняться указанные выше условия.

Управление системой для системных администраторов (SA):

Пользователь SA отвечает за аспекты администрирования системы, не связанные с защитой системы.

К числу обязанностей пользователя SA относятся следующие обязанности:

- Добавление, удаление и обслуживание учетных записей пользователей
- Совместно с пользователем ISSO обеспечение внутренней целостности программного обеспечения системы и файловых систем
- Создание и обслуживание файловых систем. Это включает планирование разбиения диска на разделы и размера разделов, выделение ресурсов для подкачки и пользовательских каталогов, мониторинг использования файловой системы, обнаружение плохих блоков и их исправление, а также обслуживание файловой системы - перемещение, удаление, архивирование файлов и каталогов.
- Обнаружение и отчет о неполадках системы посредством анализа ошибок и тестирование компонентов системы, таких как файловые системы, память и устройства.

Управление учетными записями пользователей:

Пользователь SA отвечает за добавление новых пользователей в систему. Пользователь ISSO отвечает за предоставление новым пользователям права входа в систему и выполнения команд в системе.

Дополнительная информация о предоставлении прав доступа учетным приведена в разделе записям пользователей Управление системой для Information System Security Officer.

Как только пользователь SA добавляет нового пользователя в систему, об этом следует известить пользователя ISSO, чтобы новому пользователю можно было присвоить пароль для входа в систему.

Если пользователь не должен больше работать в системе, его следует немедленно удалить. Удаление пользователя выполняет SA. ИД удаленного пользователя не должен более использоваться в системе, если не планируется восстановление исходного пользователя.

За дополнительной информацией по настройке учетных записей пользователей обратитесь к разделу к описаниям команд **mkuser**, **rmuser**, **chuser** и **pwadm**.

Работа с принтерами:

После установки принтера он добавляется в систему в результате совместных действий пользователей с правами SA и SO. Пользователь SO добавляет принтер в систему, а пользователь SA устанавливает диапазон SL принтера. У пользователя ISSO есть права на выполнение обеих этих задач.

Диапазон SL принтера должен устанавливаться только после добавления принтера в систему. Для работы с принтерами воспользуйтесь командой **smit**.

Примечание: Печать файлов PostScript и ASCII с применением меток поддерживается только на принтерах PostScript.

Доступ MAC к принтеру определяется SL процесса, печатающего файл. Эта SL указана на начальной и конечной страницах и верхнем и нижнем колонтитулах. У процесса, применяющего команду **lp**, должны быть права доступа MAC, MIC и DAC к печатаемому файлу. В противном случае команда **lp** не сгенерирует запрос на печать.

При удалении принтера из системы его профайл следует немедленно удалить. Это может сделать только пользователь с правами SO.

Управление файловыми системами:

Файловая система состоит из каталогов, файлов данных, исполняемых файлов и специальных файлов. Файловая система может находиться на различных запоминающих устройствах, таких как жесткие диски и дискеты.

Хотя создавать и обслуживать файловые системы может только пользователь с правами SA, монтировать и размонтировать файловые системы могут пользователи с правами SA и SO.

Проверка файловых систем с помощью команды fsck:

Внутреннюю целостность файловой системы следует периодически проверять с помощью команды **fsck**. Команду **fsck** следует выполнять в размонтированных файловых системах. Команду **fsck** может выполнять только пользователь с правами SA.

По умолчанию команда **fsck** выполняется интерактивно, запрашивая у пользователя действие при обнаружении неприсвоенного файла или каталога. Пользователь может удалить файл или попытаться восстановить его. Если пользователь выбирает восстановление файла, то команда **fsck** пытается сохранить файл в каталоге `/lost+found`.

После окончания выполнения команды **fsck** и сохранения восстановленных файлов в каталоге `/lost+found` пользователь с правами ISSO должен просмотреть эти файлы и определить их уровень защиты. Каталогу `/lost+found` рекомендуется присвоить уровень защиты **SYSTEM_HIGH**, чтобы восстановленные файлы были недоступны обычным пользователям.

Дополнительная информация приведена в описании команды **fsck**.

Управление системой для System Officer (SO):

Пользователь SO отвечает за аспекты администрирования системы, связанные с защитой системы.

Управление файловыми системами:

Управление файловыми системами относится к обязанностям System Officer (SO)

Поддерживаемые файловые системы:

Trusted AIX поддерживает все дисковые файловые системы.

За исключением JFS2, все файловые системы поддерживаются в Trusted AIX как одноуровневые файловые системы. Эти файловые системы можно смонтировать в системе Trusted AIX, при этом файлы автоматически получат метки и прочие атрибуты защиты, и для них будут действовать механизмы защиты Trusted AIX. Все файловые объекты в одноуровневой файловой системе имеют одинаковые атрибуты защиты. Эти атрибуты защиты наследуются от точки монтирования.

JFS2 реализована в Trusted AIX как многоуровневая файловая система. Каждый файловый объект в многоуровневой файловой системе имеет собственные атрибуты защиты (метки защиты). Например, каталог JFS2 имеет независимые минимальные и максимальные SL.

В одноуровневой файловой системе минимальные и максимальные SL точки монтирования равны, и они применяются для всех каталогов и файлов в иерархии ниже точки монтирования.

Монтирование и размонтирование файловой системы:

Пользователь SO (с правами **aix.fs.manage.mount**) может смонтировать или размонтировать файловую систему. Опциями команды **mount** являются имя устройства специального файла и каталог монтирования.

При монтировании многоуровневых файловых систем JFS2 каталогу монтирования присваивается метка корня файловой системы. В многоуровневой файловой системе каждый файл имеет собственные метки секретности и целостности. При изменении файла эти метки обновляются соответственно.

Управление принтерами:

Пользователь SO может использовать команду **lpadmin** для добавления, удаления и изменения принтеров, а также выполнять некоторые другие действия по управлению подсистемой принтеров. Пользователь SA может использовать команду **lpadmin** для добавления или изменения меток секретности (SL) для принтера и включать или выключать принтеры командами enable и disable.

Подсистема принтеров:

Подсистема принтеров обеспечивает выполнение задач, связанных с принтерами.

К задачам подсистемы принтеров относятся следующие задачи:
- Администрирование принтеров и их атрибутов
- Получение, сохранение и планирование заданий печати
- Планирование заданий печати для нескольких принтеров
- Запуск интерфейсных программ принтеров
- Отслеживание состояния принтеров и заданий
- Отправка отчетов о возникающих неполадках
- Ограничение прав доступа к заданиям принтера в соответствии с диапазоном SL принтера
- Ограничение доступа к обслуживаемым заданиям печати
- Ограничение доступа к служебным файлам и каталогам принтера
- Разметка вывода принтера

Функции защиты принтера:

Подсистема принтера Trusted AIX включает в себя некоторые дополнительные функции защиты.

Подсистема печати - это защищенная подсистема, которой владеет системный ИД **lp**. Обычные пользователи не имеют доступа к служебным файлам и каталогам принтера, кроме собственных заданий печати, или к специальным файлам устройства принтера.

Подсистема принтера проверяет, находится ли задание печати, отправленное пользователем, в пределах допустимого диапазона SL принтера. Эта проверка выполняется после отправки задания печати командой **lp** и перед выводом задания на печать демоном **lpsched**. Администратор должен знать о проверках защиты подсистемы принтера, если задание пользователя не принимается.

Для всех заданий печати выводятся титульные страницы. Титульная страница включает SL задания печати в текстовом виде. Титульная страница печатается перед каждым заданием печати и после него. Пользователь

может выключить печать титульной страницы, но это действие контролируется. Убедитесь, что метки верхнего и нижнего колонтитулов на каждой странице указаны верно, и что они поглощаются метками на титульной странице.

Примечание: Администратор принтеров должен задать диапазон меток для каждого из принтеров. Для того чтобы присвоить уникальную метку принтеру, выполните следующую команду:

lpadmin -d *принтер* **-J***метка* **-L***метка* При этом будет обеспечена печать только информации с указанной *меткой* на этом принтере.

Обзор команд работы с принтером:

Некоторые команды для работы с принтером может выполнять любой пользователь. Часть команд может выполнять только пользователь SO, SA или ISSO.

В следующей таблице перечислены команды для работы с принтером, которые может выполнять любой пользователь:

lp Отправляет файл на принтер

lpstat Показывает состояние подсистемы принтера

Администрирование подсистемы принтера выполняется с правами SO. Исключением являются команды **lpadmin**, с помощью которой можно указать диапазон меток для принтера, и **lpstat**, применяемая для показа SL принтера и заданий печати, которые можно выполнять с правами SA или ISSO. В следующей таблице перечислены административные команды для работы с подсистемой принтера:

accept Разрешает задания на принтере

cancel Отменяет запрос на печать файла

disable Деактивирует принтер

enable Активирует принтер

lpadmin
Задает или изменяет конфигурацию принтера

lpfilter Задает или изменяет фильтр принтера

lpforms
Задает или изменяет форму печати

lpmove
Перемещает запросы печати

lpsched
Печатает запрос

lpshut Останавливает службу печати

lpusers
Задает или изменяет приоритет печати

reject Запрещает задания на принтере

Управление принтером из командной строки:

Команды **accept**, **enable**, **disable**, **lpstat** и **lp** предназначены для управления принтером из командной строки.

Для отправки заданий на печать используется команда **accept**. Для того чтобы принтер *laser* начал выполнение заданий печати, используйте команду:

```
/usr/sbin/accept laser
```

Теперь принтер *laser* сможет принимать задания печати. Однако эти задания будут напечатаны только после того, как принтер будет включен. Для этого используется команда enable:

```
/usr/bin/enable laser
```

Команды **enable** и **disable** входят в число административных команд, и могут выполняться только с правами ISSO или SA.

Для проверки настройки принтера применяется команда **lpstat**:

```
lpstat -p laser -l
```

Эта команда выводит подробный отчет о состоянии принтера *laser*. Команда **lpstat** без опции **-l** выводит краткий отчет. Если эту команду вызывает пользователь с правами SA или ISSO, и используется опция **-l**, то также показывается диапазон SL принтера.

Для определения состояния задания принтера применяется следующая команда **lpstat**:

```
lpstat -o
```

Эта команда выводит список всех заданий печати **lp**. Если ее выполняет пользователь с правами SA или ISSO, то будет показан также фактический SL и допуск каждого задания.

Для печати файла используйте команду **lp**:

```
lp -d laser файл
```

В противном случае необходимо указать целевой объект задания печати при вызове команды **lp**.

если администратор настроил принтер по умолчанию, то опция **-d** *целевой-объект* необязательна. Например, для печати файла на принтере laser выполните следующую команду **lp**:

```
lp файл
```

Управление выключением системы:

Пользователь SO может завершить работу системы, либо перезагрузив ее, либо полностью выключив ее.

Следующие команды применяются пользователем SO для перезагрузки или выключения питания системы, а также для изменения уровня init системы:

reboot Автоматически перезагружает систему

halt Останавливает все операции в системе

shutdown
Останавливает все операции в системе

init Изменяет уровень init системы

Резервное копирование и восстановление файлов:

Резервное копирование позволяет предотвратить потерю данных при отказах аппаратного обеспечения или случайном удалении файла. Резервное копирование должно выполняться регулярно, при этом в промежутках между полным резервным копирование должно выполняться дополняющее резервное копирование.

В командах **backup** и **restore** предусмотрены опции для указания имени файла резервной копии, расположения, типа и т.д. Командой **mksysb** можно создать установочный образ корневой группы томов Trusted AIX в файле или на магнитной ленте. Эти команды можно выполнить с помощью команды **smit**. Резервные копии файловой системы должны быть описаны надлежащим образом и должны храниться в надежном месте.

Программирование Trusted AIX

Защита системы системы зависит от таких компонентов защищенной компьютерной базы (TCB), как программное обеспечение, аппаратное обеспечение и встроенное ПО. В это входит полностью ядро операционной системы, все драйверы устройств и модули System V STREAMS, расширения ядра и все защищенные программы. Все файлы, к которым обращаются эти программы при принятии решений, связанных с защитой, также являются частью TCB.

Создание защищенного программного обеспечения требует глубокого понимания принципов защиты системы. Почти все изъяны защиты в UNIX-подобных системах обусловлены ошибками в написании защищенного программного обеспечения. Однако с помощью проверок защиты ядра Trusted AIX можно создавать приложения, использующие улучшенные функции защиты. Приложение для Trusted AIX может избирательно работать с файлами и процессами на разных уровнях защиты, и может вести себя по-разному в зависимости от уровня процесса или файла, с которым оно работает. Такие приложения называются приложениями с многоуровневой защитой (MLS).

Программист должен в совершенстве овладеть работой с функциями защиты Trusted AIX и должен знать все новые системные вызовы Trusted AIX и связанные с защитой команды и библиотеки. Эта информация предназначена для программистов, которые создают или изменяют защищенное программное обеспечение. Приведены рекомендации, принципы и советы по написанию защищенного программного обеспечения. Этот материал может служить введением в принципы и подходы работы с защитой, и программистам рекомендуется прочитать и другие руководства по защищенным системам.

Принципы надежного программного обеспечения

Создание и модификация надежного программного обеспечения основываются на нескольких важных принципах, включая надежность и привилегии, проектирование надежного программного обеспечения, принцип наименьших привилегий, соглашения о программировании и защита TCB.

Защищенные и привилегированные процессы:

Процесс может обойти основные ограничения защиты (MAC, MIC, DAC и прочие), только если ему предоставлены соответствующие права доступа. Любой процесс, работающий с повышенными правами доступа, называется привилегированным процессом, а программа, запустившая процесс, называется привилегированной (или защищенной) программой.

Термин привилегии означает индивидуальный атрибут, разрешающий процессу выполнять операции, связанные с защитой системы. В Trusted AIX выделены группы операций, связанных с защитой, и с такими операциями связаны разные уровни прав доступа. При этом права доступа суперадминистратора (root) фактически удаляются из базовой системы. Права доступа связываются с процессами и исполняемыми файлами.

Программы необходимо сделать привилегированными в следующих ситуациях:

- Программа настроена или будет выполняться как привилегированный процесс. Это относится ко всем программам, которые будут выполняться привилегированными процессами.
- К этой программе будет обращаться другая защищенная программа при принятии решений, связанных с защитой. Например, если программа изменяет данные в базе данных, связанные с защитой, то она должна быть защищенной, если другие программы принимают решения на основе данных из этой базы данных.

Важно обеспечить невозможность выполнения незащищенных программ как привилегированных процессов. Для этого существует несколько способов:

- Нельзя разрешать в обычных условиях защищенным программам запускать незащищенные программы. Например, следует запретить пользователям, работающим в привилегированной оболочке, запускать в ней незащищенные программы.
- Никогда не следует предоставлять незащищенным исполняемым файлам встроенные, унаследованные или повышенные права доступа.

Все части ядра операционной системы, включая драйверы устройств, модули STREAMS и расширения ядра, должны быть защищенными. Объекты данных, такие как файлы и физические устройства, также должны считаться защищенными, если они содержат данные, которые используются защищенной программой в принятии решений, связанных с защитой.

Проектирование защищенного программного обеспечения:

Процесс создания защищенного программного обеспечения аналогичен процессу создания любого критически важного программного компонента. В ходе создания защищенного программного обеспечения необходимо в точности следовать документации, в которой описаны этапы разработки: создание спецификации, проектирование, реализация, тестирование и проверка конфигурации.

Наиболее важными аспектами в проектировании защищенного программного обеспечения являются описание субъектов и объектов и точное определение действий, связанных с безопасностью, на соответствующем уровне абстракции. Большинство стратегий защиты применяются как ограничения для субъектов, объектов и действий. Когда субъект запрашивает разрешение на чтение, изменение или создание объекта, монитор стратегии защиты оценивает этот запрос и разрешает или запрещает его выполнение.

Субъекты

Субъекты обычно представляются как ИД пользователей или групп. При этом используется эффективный ИД пользователя и/или группы, но в некоторых ситуациях может применяться и фактический ИД пользователя и/или группы.

Объекты

Объектом является любой набор данных, доступом к которому необходимо управлять. В большинстве случаев объекты - это файлы. Даже если защищенная программа часто управляет доступом к разным объектам в одном и том же файле, более правильным способом будет взаимно однозначное отображение объектов на файлы.

Иногда субъект также может выступать как объект. Например, процесс обычно считается субъектом. Однако если этот процесс воздействует на другой процесс, то в этой операции второй процесс будет считаться объектом.

Запросы

Запросы - это наборы действий, которые защищенный модуль может выполнять от имени субъекта. Каждый запрос должен иметь четко определенные входы запроса, возможные выходы и результаты, включая и все побочные эффекты. Строгое обозначение всех запросов является важным предварительным фактором в определении стратегии защиты.

Стратегии защиты

Стратегии защиты строятся на основе простых правил, указывающих, как запросы с указанными объектами будут выполняться от имени указанных субъектов. Субъекты, объекты и запросы должны быть точно определены, а стратегии защиты должны быть сформулированы четко и кратко. Для целей контроля важно указать идентификацию субъекта запроса и связанных объектов.

Принцип наименьших привилегий:

Принцип наименьших привилегий гласит, что модулям программного обеспечения должны предоставляться минимальные возможности, необходимые для выполнения поставленной перед ними задачи.

Принцип наименьших привилегий означает также, что надежные программы должны добровольно ограничивать свои возможности по работе с секретной информацией, так чтобы они использовались лишь в

тех областях программы, где они действительно необходимы. Принцип наименьших привилегий помогает снизить ущерб от ошибок программного обеспечения и непредвиденных побочных эффектов. Все надежное программное обеспечение должно разрабатываться согласно принципу наименьших привилегий.

Присвоение и удаление прав доступа:

Один из принципов работы защищенного программного обеспечения заключается в том, чтобы все действия, для которых необходимы права доступа, производились на раннем этапе выполнения программы, а затем, на протяжении остального времени работы программы, проверка прав доступа снимается. Это называется заключением прав доступа в скобки.

В связи с использованием прав доступа, следует принять во внимание следующее:

- Каждому пользовательскому процессу присваивается набор максимальных прав доступа при выполнении процесса. Этот набор прав доступа всегда можно сократить, но расширить его может только администратор.
- При выполнении операций, для которых необходимы определенные права доступа, на процесс выполнения возлагается ответственность за повышение и понижение прав доступа из максимального набора, путем помещения прав доступа в эффективный набор и удаления их из него.
- Права доступа процесса меняются при запуске процессом выполняемых файлов, имеющих непустые исходные наборы прав доступа. Дополнительная информация приведена в описании команды **exec**.
- При запуске процессов им также присваивается ограничивающий набор прав доступа. При наличии соответствующих прав доступа, процесс может повысить права доступа из максимального набора до уровня прав доступа ограничивающего набора.

Быстрое изменение меток MAC:

Операцию изменения метки MAC процесса необходимо завершить как можно быстрее. Это достигается благодаря использованию библиотечных процедур.

Дополнительная информация об этих библиотечных процедурах приведена в разделе “Системные вызовы защищенной AIX” на стр. 499.

Быстрое закрытие критических файлов:

Некоторые файлы содержат информацию, критически важную для защиты системы, например, теневой файл с паролями. Эти файлы следует открывать для чтения или записи на минимально короткое время.

Необходимо задать атрибут **close-on-exec** дескриптора файла с помощью системного вызова **fcntl**. Это не позволяет несанкционированным процессам наследовать дескриптор открытого файла посредством системного вызова **exec**.

Централизация секретных операций:

Секретной называется операция, для выполнения которой необходимы определенные привилегии. Если секретная операция выполняется процессом, не имеющим таких привилегий, то это может нарушить защиту системы.

Секретные операции должны выполняться только самостоятельными модулями (подпроцедурами или отдельными программами). Если большая программа разбита на отдельные программы, то некоторым из них понадобятся меньшие привилегии или не понадобятся никакие. Это снижает вероятность случайного нарушения защиты системы.

Использование эффективных корневых каталогов:

Программу можно ограничить в каком-либо каталоге, сделав этот каталог эффективным корневым каталогом командой **chroot**, и указав рабочий каталог программы также внутри этого каталога. По сути это есть реализация принципа наименьших прав доступа, поскольку при этом ограничивается область файлов, к которым имеет доступ процесс, даже если это привилегированный процесс. При этом также фактически ограничивается область доступа дочерних защищенных или незащищенных процессов, которые запускает родительский процесс.

Несмотря на то, что смена корневого каталога защищает файлы вне этого каталога, возникает другая потенциальная проблема с защитой. При невнимательном подходе к смене корневого каталога защита нового корневого каталога может быть нарушена. Это может иметь место, если возможна подмена динамического компоновщика и общих объектов в новом корневом каталоге. Таким образом, эта процедура должна использоваться с осторожностью.

Использование защищенных подсистем:

Защищенные подсистемы обеспечивают целостность для специальных подсистем. Подсистема - это набор программ и/или файлов данных, которые принадлежат одному и тому ИД пользователя и/или ИД группы и применяются для реализации какой-либо функции в системе.

Подсистема может включать программы setuid или setgid. Защищенная подсистема - это подсистема, которая принадлежит системному ИД.

Системные ИД имеют численное значение, меньшее или равное 127. Пользователи не могут входить в систему с системными ИД. Применение защищенных подсистем может заметно снизить число привилегированных процессов.

Режимы минимально необходимого доступа:

Надежные программы (фактически это все программы) должны открывать объекты только в абсолютно необходимых им режимах чтения и записи. В общем случае, это означает, что объект не следует открывать для чтения и записи, если достаточно открыть его только для чтения. В случае особых требований к защите, процесс должен открывать объект для записи только там, где необходимо ее выполнить.

Все эти рекомендации особенно важны, если программа создает другие процессы, поскольку передача привилегий и прочие функции общего назначения (например, открытые соединения с секретными файлами) занимают узловое место в проектировании надежного программного обеспечения. Привилегии могут переопределять все ограничения. При создании новых команд, у которых будут привилегии, необходимо тщательное проектирование с учетом вышеописанных требований.

Другие надежные соглашения о программировании:

Trusted AIX применяет много других надежных соглашений о программировании.

Избыточность:

Избыточность - это полезная концепция для систем защиты. Защита редко бывает абсолютной; напротив, она почти всегда напоминает воздвижение достаточного количества препятствий на пути тех, кто пытается получить несанкционированный доступ к системе.

Преимущество избыточных мер защиты заключается в том, что если одна из них даст сбой или окажется нарушенной, то остальные обеспечат необходимую защиту. Недостаток - в том, что общие меры защиты оказываются разделенными или распределенными в системе. Таким образом, хотя избыточные меры могут оказаться исключительно полезными, их следует тщательно проектировать, документировать и обслуживать.

Предпочтительное выполнение проверок ядром:

Как правило, процессу не рекомендуется выполнять проверку, которую может выполнить ядро. Например, процессу не следует считывать метку MAC файла и самому выполнять проверку MAC. Всегда, когда это возможно, проверку следует поручать ядру.

Ядро должно выполнять проверки по следующим двум основным причинам.

- Операции ядра атомарны по отношению к другим процессам, в то время как проверки, выполняемые процессами, могут конкурировать с другими процессами.
- Еще важнее, что применяемые алгоритмы могут изменяться с каждой новой версией ядра. Отслеживать такие изменения в алгоритмах, входящих в программное обеспечение конечного пользователя, нелегко.

Прямая проверка привилегированности:

Программы не должны пытаться определять, запускаются ли они как привилегированные процессы (например, выясняя свой действующий или максимальный вектор привилегий). Вместо этого, программы должны заранее считать, что они запускаются как привилегированные, там, где это необходимо.

Если программа не является привилегированным процессом, то привилегированные системные вызовы выполняться не будут и программа сможет выполнить соответствующее действие. Как правило, самостоятельный отказ программы от выполнения тех или иных операций в непривилегированном режиме не является эффективной мерой защиты. Если программа привилегированная, то проверка бессмысленна. Если программа непривилегированная, то она может причинить не больше вреда, чем любой другой непривилегированный процесс.

Однако такая проверка может помочь при непреднамеренном неправильном использовании. Может выдаваться осмысленное сообщение об ошибке, указывающее, что программа должна быть привилегированной, но не является таковой.

Распространение секретных возможностей:

Секретной возможностью называется функция надежной программы, которая в случае ее предоставления ненадежной программе может нарушить защиту системы.

Когда привилегированная программа передает свои привилегии или общие функции другим программам через семейство **fork** и **exec** системных вызовов, необходимо соблюдать осторожность. Системные вызовы **exec** наиболее важны, поскольку они передают привилегии от одной программы другой. Системный вызов **fork** создает новый процесс, но с теми же привилегиями, что и у родительского процесса. Основная опасность заключается в том, что исполняемый файл программы может быть ненадежным или мог быть изменен ненадежной программой. Необходимо принять во внимание следующее:

- Надежные программы не должны передавать открытые соединения с объектами (преимущественно файлами) дочернему процессу, кроме случая, когда дочернему процессу и его потомкам можно доверять, когда они обращаются к файлу в режиме, в котором он был открыт. Лучше всего, если процесс передаст новое соединение с объектом, режимы которого более ограничительны, чем ожидаемые.
- Надежный процесс, работающий с действующим корневым каталогом, отличным от абсолютного корня, должен быть уверен, что его дочерний процесс не запутается. Например, когда дочерняя программа открывает надежный файл, такой как теневой файл паролей, она может использовать абсолютный путь в предположении, что его действующий корневой каталог является абсолютным.
- Возможны случаи, когда надежной программе требуется наложить более ограничительное значение umask на свои потомки.
- Дочерние процессы наследуют многие атрибуты процесса. Если надежной программе известно, что дочерний процесс ненадежен, его метка MAC не поглощает метку надежного процесса и эти атрибуты были унаследованы надежной программой от ненадежного предка, то эти атрибуты могут послужить причиной возникновения потенциальных скрытых каналов.

- Ознакомьтесь с правилами распространения привилегий для системных вызовов **fork** и **exec**. Когда происходит системный вызов **fork**, привилегии родительского процесса становятся привилегиями дочернего процесса. Во время системного вызова **exec** привилегии изменяются.

В ситуациях с исключительно высокими требованиями к секретности надежная программа может изучить управление доступом надежного файла и убедиться, что файл правильно защищен от изменения ненадежными программами. Например, можно потребовать, чтобы файл принадлежал пользователю root, с предоставлением прав не более чем на запись DAC владельцу файла.

Среды действующего root:

Надежные программы часто полагаются на правильные абсолютные пути. Например, программа **login** опирается на файл `/etc/security/passwd`, который является правильным теневым файлом паролей.

Надежные программы используют не только файлы данных, но и исполняемые файлы. Хотя ненадежная программа не может воспользоваться системным вызовом **chroot** для непосредственного изменения действующего корневого каталога программы, возможны ситуации, когда TCB разрешает ненадежным программам работать с действующим root. Если эти ненадежные программы могут выполнять надежную программу, опирающуюся на абсолютный путь, то это может вызвать нарушение защиты.

Идентификация с помощью фактических и действующих идентификаторов:

Защищенные программы могут требовать использования нескольких идентификаторов пользователей и групп, связанных с процессом. Важно понимать различия между этими идентификаторами и их соответствующее использование.

Фактические идентификаторы пользователей и групп

Фактические идентификаторы пользователей и групп, как правило, представляют имя и сеанс пользователя, в котором был создан процесс. В некоторых случаях фактические идентификаторы (в частности, фактический ИД пользователя) можно использовать в решениях защиты. Одним из таких случаев является проверка прав доступа. Фактические идентификаторы пользователя используются командами как форма проверки субъекта. Это особенно полезно для пресечения злонамеренного или беспечного использования управляющих разрядов **setuid-on-exec** или **setgid-on-exec**. Вместе с тем, проверка фактических ИД отступает от общепринятой практики UNIX, и ее следует выполнять лишь по необходимости. В системах UNIX действует общий принцип: действующие ИД используются для проверки доступа и других связанных проверок защиты. Не следует отступать от этой принятой практики без глубокого обдумывания и документирования.

Действующие идентификаторы пользователей и групп

Действующие ИД пользователей и групп следует использовать во всех решениях об управлении правами доступа (DAC и MAC). Значения ИД пользователей системы находятся в диапазоне от 0 до 127. Значения ИД обычных пользователей начинаются со 128.

Полные пути для защищенных команд:

Некоторые схемы преодоления защиты основаны на создании подделки под защищенную программу, которая помещается в программу-оболочку, используемую администратором или даже обычным пользователем. Например, для несанкционированного копирования пароля существующего или нового пользователя может быть использована фальсифицированная копия команды **passwd**.

На практике, в целях эффективной защиты от подобных вторжений, администратор помещает текущий рабочий каталог вне пути поиска. Тем не менее, могут существовать и другие пути поиска, не всегда надежно защищенные, и обычным пользователям должно быть разрешено размещать текущие рабочие каталоги в своих путях поиска. Действенной контрмерой является введение правила, согласно которому вызвать

защищенную программу можно только через полный путь /usr/bin/passwd). Защищенная программа сама проверяет первый аргумент запуска и имя запуска. Если соответствующий полный путь не используется, защищенная программа запущена не будет. Кроме того, защищенная программа проверяет отсутствие фактического корневого каталога, отличающегося от полного корневого пути.

Примечание: Данная мера является эффективной лишь при условии, что пользователи будут надлежащим образом обучены и всегда будут вызывать программу через полный путь. Если пользователь по неосторожности воспользуется относительным путем, вместо полного, и запустит таким образом программу-подделку, возможно срабатывание схемы преодоления защиты.

Структурирование дерева каталогов:

Дерево каталогов следует тщательно структурировать, чтобы усилить защиту важнейших файлов. Основная рекомендация заключается в том, что доступ для поиска в каталоге должен быть максимально ограничивающим (например, все общедоступные файлы рекомендуется помещать в каталоги, близкие к корню файловой системы).

Кроме того, имеет смысл помещать важнейшие секретные каталоги максимально близко к абсолютному корню, чтобы минимизировать число промежуточных каталогов, требующих защиты.

Файловые системы, доступные только для чтения:

Интересной особенностью дерева каталогов является возможность размещения редко изменяемых надежных файлов в их собственной файловой системе и их монтирования как доступных только для чтения. Это, в сущности, означает, что их содержимое нельзя изменить в рамках обычной работы системы. Этот прием часто используется в больших совокупностях исполняемых файлов, предназначенных для надежных программ.

Если необходимо изменить файл, то файловую систему можно заново смонтировать как доступную для записи в более защищенном контексте (например, в однопользовательском режиме или на отдельном, лучше защищенном компьютере). Рекомендуется после таких обновлений проверять правильность конфигурации (например, правильность меток DAC, MIC и MAC) файловой системы с помощью соответствующих программ.

Кроме того, информацию DAC, MIC и MAC нельзя изменить в файловой системе, предназначенной только для чтения. Если файловая система настроена правильно, то это должно защитить от схем нарушения защиты, пытающихся изменить информацию DAC и/или метки MIC и MAC.

Обработка паролей:

Как правило, программам, за исключением стандартных системных утилит, не рекомендуется запрашивать у пользователя пароль входа в систему. Пароли - это исключительно секретная информация, и их обработка должна быть строгой прерогативой нескольких абсолютно надежных системных утилит.

В некоторых надежных подсистемах может оказаться удобным реализовать собственные пароли. Однако полагаться на такие частные схемы реализации паролей рискованно, поскольку они не так безопасны, как системные механизмы.

Безопасность защищенной компьютерной базы (TCB):

Файлы, содержащие элементы TCB, должны быть защищены от изменения, а в некоторых случаях и от чтения несанкционированными программами.

Защита от изменения критически важна. Защита от чтения также может быть критически важна. Указанные далее файлы должны быть защищены:

- Все файлы, которые используются защищенной программой для операций с защитой (например, теневой файл паролей)
- Все исполняемые файлы защищенных программ
- Псевдофайлы, обеспечивающие доступ к компонентам TCB (например, `/dev/kmem`).

Примечание: Файлы инициализации системы (файлы rc) также являются частью TCB и требуют особой защиты

Защита от модификации:

Защита от несанкционированной модификации достигается прежде всего за счет присвоения соответствующего значения информации DAC. В обычной ситуации эти файлы будут принадлежать системному ИД пользователя, с предоставлением прав на запись только владельцу файла.

MIC защищает от модификации путем обеспечения целостности объектов. Если присвоить файлу высокую метку MIC, то процессы с меньшей меткой MIC не смогут изменить, удалить или переименовать файл. Это идеальный способ предотвратить нежелательное изменение файлов.

В некоторых случаях для защиты от несанкционированной модификации можно воспользоваться MAC. Однако MAC предназначен для защиты от считывания и не слишком подходит для защиты от модификации. Основная стратегия MAC не запрещает субъектам модифицировать объекты с более высокими метками. Хотя это и не разрешено в форме прямой записи в файлы, некоторые надежные подсистемы могут допускать это. Кроме того, многим надежным файлам, например исполняемым файлам программ, необходимо присваивать низкую метку MAC, чтобы они были, вообще говоря, доступны. Таким образом, присвоение файлу высокой метки MAC не всегда приемлемо.

Флаги защиты файлов также защищают файлы от модификации. Некоторые флаги защиты файлов запрещают модификацию объектов даже привилегированными субъектами. Если для файла задан флаг защиты **FSF_TLIB**, то файл можно изменять, только когда система находится в режиме настройки, при условии, что включен флаг защиты ядра **trustedlib_enabled**. Для того чтобы процесс мог задать флаг **FSF_TLIB** для файла, в EPS процесса должна существовать привилегия **PV_TCB**. Другим подходящим флагом защиты файла является **FSF_APPEND**, предотвращающий модификацию ранее записанных данных. В файл с заданным флагом **FSF_APPEND** можно лишь добавлять данные. Это может оказаться полезным для приложения, заносящего записи в файл.

Эти флаги обычно задаются для файлов интеграторами, а не программами. Программистам должны быть известны эти флаги и их функции.

Защита от считывания:

С помощью DAC и MAC можно защитить файлы TCB от чтения. Метки MAC в этих файлах должны в точности отражать секретность информации в этих файлах. Например, если какой-либо алгоритм секретен, то метка MAC в исполняемом файле программы, использующей этот алгоритм, должна быть задана соответствующим образом.

Общепринятой практикой является установка достаточно высокой метки MAC (выше фактического уровня секретности данных в файле) для защиты данных от считывания. Однако такими завышенными уровнями секретности не следует злоупотреблять.

Практически во всех случаях для обеспечения адекватной защиты файла необходимо защитить всю цепочку каталогов начиная от абсолютного корня. В противном случае, вредоносная программа сможет удалить незащищенную часть цепочки каталогов и создать новое поддерево с поддельной копией файла.

Предположим, например, что полный путь к надежному файлу имеет вид `/A/B/foo`. В то время как файл **foo** защищен от изменения, каталог **B** - нет. Вредоносная ненадежная программа сможет удалить ссылку в **B** на

foo и создать новый файл **foo** с фальшивой копией старого файла **foo**. В результате этого надежные программы, открывающие `/A/B/foo`, будут открывать фальшивый файл и получать неверные данные.

При обращении к файлам TCB надежные программы полагаются на правильные пути. По этой причине, файлы символьных ссылок, используемые в путях к файлам TCB, должны быть защищены так же надежно, как и сами файлы.

В некоторых случаях для защиты от несанкционированного считывания можно воспользоваться MIC. Однако MIC предназначен прежде всего для защиты от модификации (записи) и не слишком подходит для защиты от считывания.

Операции с метками секретности:

Существуют рекомендации для защищенных программ по работе субъектов и объектов с различными уровнями меток секретности.

Необходимо ознакомиться с форматом меток секретности и с отношением поглощения между метками. Поглощение означает более высокий уровень метки, поглощаемая метка имеет более низкий уровень. Повышение уровня означает, что категория данных метки поднимается до более высокого уровня, а понижение уровня - что она понижается до более низкого уровня.

Основное ограничение MAC:

Основное ограничение в мандатном контроле за доступом состоит в том, что незащищенные субъекты не могут вызвать изменение метки чувствительности данных, помеченных A, на B, если B не доминирует над A.

Основное ограничение MAC относится ко всем классам данных. Оно включает ограничения на смену меток данных (то есть, смену меток контейнера данных) и на перемещение помеченных метками данных между контейнерами данных.

На разных уровнях системы (системный вызов, служебные утилиты системы и т.д.)это основное ограничение принимает форму более конкретных наборов правил, но непременно на основе одного базового принципа - данные можно только модернизировать. Например, на первом уровне расширения процессы могут открывать для чтения любой крупный класс объектов, если метка процесса доминирует над меткой объекта, и открывать для записи, если метка объекта доминирует над меткой процесса.

Для обычного файла операции записи имеют следующее ограничение: они возможны только для файлов с той же меткой, что и процесс. Для каталогов и устройств операции записи разрешены, если SL субъекта доминирует над минимальной SL объекта, и максимальная SL объекта доминирует над SL субъекта. Для особых файлов FIFO (конвейеры) операции чтения также ограничены особыми файлами FIFO с той же меткой, что и процесс, по соображениям скрытых каналов.

Несмотря на то, что данные можно переносить к меткам более высокой чувствительности, такая возможность не является обязательной для конкретного объекта и ситуации. Например, операционная система сама по себе не допускает, чтобы непривилегированный процесс открывал для записи файл с более высокой меткой, несмотря на то, что в соответствии с основным ограничением MAC это допустимо. Вопрос о возможности такой модернизации для незащищенных субъектов решается на уровне проектирования и стратегии. В одних случаях это полезно, в других - нет. Например, трудности, связанные с прямой записью в файлы с более высокой меткой, заключаются в том, что процессу не удается прочитать эти файлы, и потому вопрос о пользе записи в файлы с более высокой меткой не стоит. В то же время, простая защищенная утилита, которая подняла метку файла по запросу от незащищенного субъекта, может быть допустимой и полезной утилитой.

На уровне системных вызовов ограничение накладывается только на непривилегированные процессы. А привилегированные процессы не связаны таким ограничением. Однако практически все службы,

выполняемые защищенной системой, будут разработаны для незащищенных пользователей, и потому на уровне пользовательских служб ограничение действует.

Основное ограничение MAC относится ко всем средствам передачи данных, которые имеются в распоряжении незащищенных программ. Однако основное ограничение MAC часто подразделяется на два компонента. Первый компонент касается только тех функций операционной системы, которые предназначены для передачи данных (или присваивания меток). К таким функциям относятся, например, чтение и запись файлов и обмен данными между процессами. Второй компонент касается средств связи, не предназначенных для этих целей; они носят название скрытых каналов. В отношении скрытых каналов практически невозможно принудительно выполнить основное ограничение MAC. Поэтому допускается существование скрытых каналов с низкой скоростью передачи данных (например, 0,1 бит в секунду), но только в тех случаях, когда есть разумный компромисс за счет других факторов.

Основное ограничение MAC является прямым и простым, и имеется сравнительно мало подробных инструкций, касающихся работы с многоуровневыми данными.

Многоуровневые операции:

Системный вызов **sec_setplab** позволяет привилегированному процессу произвольно изменять свою метку.

Поскольку почти все ограничения MAC и MIC в непривилегированных процессах применяются и к привилегированным процессам в стандартных системных вызовах (т.е. тех, которые определены в базовой операционной системе), то привилегированные процессы, которым необходимо выполнять многоуровневые операции, должны полагаться в основном на системный вызов **sec_setplab**. Однако надежные программы должны использовать sec_setplab() только следующим образом:

- Любое применение системного вызова **sec_setplab** для выполнения многоуровневых операций (например, открытия файлов с метками более высокого уровня для чтения) должно выполняться только посредством библиотечных процедур, отражающих семантику фактической выполняемой высокоуровневой операции и скрывающих детали применения системного вызова **sec_setplab**.
- Единственное исключение - очень простые изменения меток процессов, выполняемые вне многоуровневых операций. Эти простые операции могут использовать системный вызов **sec_setplab** напрямую.

Эти рекомендации по применению системного вызова **sec_setplab** основываются на двух причинах. Во-первых, такая секретная и потенциально опасная функция, как системный вызов **sec_setplab**, должна применяться только в правильно спроектированной, модульной форме. Во-вторых, по мере развития стандартов для надежных систем низкоуровневые системные вызовы могут начать поддерживать различные механизмы многоуровневых операций.

Инкапсуляция высокоуровневых операций в библиотечных процедурах обеспечивает хорошую совместимость с предыдущими версиями и приспосабливаемость к последующим версиям операционной системы, а также помогает обеспечить переносимость между надежными версиями системы UNIX.

Надежная система предоставляет базовый набор таких процедур. Эти процедуры следует использовать всегда, когда это возможно. Этот набор процедур следует расширять в последующих версиях операционной системы. Программист надежной системы может также создавать такие библиотечные процедуры при необходимости.

Другое исключение для ограничений MAC и MIC - это применение одной или нескольких доступных привилегий MAC или MIC для обхода ограничений MAC или MIC. Разрешать использование любой из этих привилегий следует с осторожностью.

Взаимодействие между процессами в System V (System V IPC):

Взаимодействие между процессами (IPC) (очереди сообщений, семафоры, общая память) также подчиняется ограничениям DAC, MIC и MAC. Как правило, не существует команд для создания объектов System V IPC и работы с ними.

В AIX системные вызовы, связанные с IPC, были изменены с учетом многоуровневой защиты Trusted AIX. Ниже они перечислены:

- **msgget**
- **msgsnd**
- **msgrcv**
- **msgctl**
- **semget**
- **semop**
- **semctl**
- **shmget**
- **shmctl**
- **shmat**
- **shmdt**

Помимо этого для работы с атрибутами MAC объектов IPC в Trusted AIX были добавлены следующие системные вызовы:

sec_getmsgsec
: Получить атрибуты защиты очередей сообщений

sec_getsemsec
: Получить атрибуты защиты семафоров

sec_getshmsec
: Получить атрибуты защиты сегментов общей памяти

sec_setmsglab
: Задать атрибуты защиты очередей сообщений

sec_setsemlab
: Задать атрибуты защиты семафоров

sec_setshmlab
: Задать атрибуты защиты сегментов общей памяти

Дополнительная информация по требованиям к правам доступа процессов, работающих с объектами IPC, приведена в разделе Доступ к объектам IPC. Для работы с атрибутами IPC можно использовать команду **settxattr**.

Высшие метки MIC и MAC реализации и системы:

Надежной программе часто бывает необходимо определить метку MAC, поглощающую все остальные метки в системе. Предусмотрено две различных метки MAC - высшая метка MAC реализации и высшая метка MAC системы.

Высшая метка MAC реализации - это наибольшая метка MAC, поддерживаемая Trusted AIX. Как правило, эта метка занимает определенное место в иерархии и содержит категории, не используемые на данном сайте. Она легко генерируется, но применять ее следует с осторожностью. Ни один процесс не должен создавать объекты с этой меткой.

Высшая метка MAC системы - это наибольшая метка MAC, используемая на данном сайте. Она определяется администратором в файле **LabelEncodings**.

Применение высшей метки MAC системы менее эффективно, но настоятельно рекомендуется, поскольку администратор может легко ограничить действия даже привилегированного процесса, правильно задав соответствующий параметр в файле **LabelEncodings**.

У MIC существуют аналогичные высшие метки реализации и системы.

Системные и пользовательские диапазоны для входа в систему:

Защищенным программам, выполняющим операции для пользователей, может потребоваться ограничить метки MIC и MAC, участвующие в этих операциях, значениями, которые допускают вход данного пользователя в систему, и/или метками, которые допускают вход в систему на уровне всей системы.

Допуски пользователей задаются в файле **user** базы данных (`/etc/security/user`), и для работы с ними применяются библиотечные процедуры **getuserattr** и **getuserattrs**.

В Trusted AIX пользователь может работать в системе с любой меткой, которая указана в разрешенном диапазоне для системы, которая поглощается максимальным допуском пользователя и которая поглощает минимальный допуск пользователя. Все программы, которые позволяют работать с другими метками, должны проверять правильность новой метки для пользователя.

Предположим, что программа **upgrade** позволяет повысить уровень метки MAC для файла по запросу любого пользователя. Основное ограничение MAC требует, чтобы **upgrade** принимала только файлы, метка MAC которых поглощается меткой пользователя. Кроме того, представляется разумным (даже если это не является обязательным условием согласно ограничениям MAC), чтобы новая метка была установлена равной метке, с которой пользователю разрешен вход в систему, согласно разрешенным диапазонам пользовательских и системных меток. Для этого программа **upgrade** может использовать интерфейсы **sl_cmp** и **accredrange**.

Структура дерева каталогов:

Системные вызовы работают таким образом, что структуры каталогов, создаваемые непривилегированными процессами, обладают метками с неуменьшающейся значимостью. Это означает, что метка файла равнозначна метке его родительского каталога или находится в диапазоне для разделенного каталога, а метка каталога поглощает метку родительского каталога (с учетом того что равнозначные метки поглощают друг друга). Это естественная структура для незащищенных программ.

Однако привилегированные процессы не связаны этим ограничением и могут создавать структуры каталогов с произвольным отношением между метками MAC каталогов. Такие конфигурации могут быть полезны, чтобы ограничить доступ при поиске MAC областью около корня дерева. Например, защита сводки, при которой метка MAC набора объектов данных имеет более высокий уровень, чем отдельная метка любого из объектов, может быть реализована тем, что метка MAC каталога получает более высокий уровень, чем метка любого из его элементов. При этом для доступа к сводке данных незащищенные процессы должны поглощать метку каталога.

Создание структур каталогов с уменьшающимся уровнем меток следует выполнять очень осторожно. Если метка файла не поглощает метку родительского объекта, то непривилегированный процесс не сможет открыть файл для записи.

Работа с разделенными каталогами:

Существует несколько системных вызовов, которые функционируют иначе в результате реализации разделенных каталогов.

Следующие системные вызовы функционируют иначе в результате реализации разделенных каталогов:

- getdirents
- link
- mkdir
- mount
- rename

- rmdir
- stat
- lstat
- fstat

Режим процесса:

Команда **pdmode** применяется для выполнения команды в заданном режиме. Процесс может использовать вызов **setppdmode** для своего перехода в реальный или виртуальный режим. Вызов **setppdmode** требует прав доступа **PV_PROC_PDMODE**. Механизм изменения режима другого процесса не предусмотрен.

Тип каталога:

Команду **pdset** можно использовать для преобразования обычного каталога в разделенный каталог. Однако команды для преобразования разделенного каталога или разделенного подкаталога в обычный каталог не предусмотрено.

Для создания разделенных каталогов применяется также системный вызов **pdmkdir**. Системный вызов **pdmkdir** требует прав доступа **PV_FS_PDMODE**.

Замечания о метках MIC и MAC:

Для определения взаимосвязей между метками MIC и MAC программы должны применять только функции **sl_cmp** и **tl_cmp**.

Причина этого состоит в том, что внутренний формат меток может изменяться в последующих версиях системы, и эти библиотечные процедуры учитывают изменения формата. Есть и прочие библиотечные процедуры, работающие с метками MIC и MAC, и следует применять именно их.

Системные вызовы **setea**, **lsetea** и **fsetea** изменяют метки MIC или MAC файла. Системный вызов **fsetea** принимает дескриптор файла.

Драйверы устройств:

Существуют определенные принципы и рекомендации, которых следует придерживаться при создании драйверов устройств для систем Trusted AIX. Вы должны быть знакомы с приемами создания драйверов устройств для основной системы и мерами предосторожности, связанными с использованием этих приемов.

Подсистема управления устройствами:

Устройство в системе AIX - это абстрактное понятие, охватывающее все объекты данных, которые используются при обращении к специальным файлам устройств. В некоторых случаях эти объекты данных представляют фактические физические устройства, а в других они весьма отличаются от них (например, в случае `/dev/null` объект данных вообще отсутствует). В последних случаях говорят о псевдоустройствах.

В системах Trusted AIX устройства бывают двух типов: с одной меткой и многоуровневые. Многоуровневое устройство безопасно обрабатывает данные на нескольких уровнях секретности одновременно. Устройство с одной меткой обычно ненадежное. Метки данных обычно связаны с информацией, которую многоуровневое устройство обрабатывает таким образом, который гарантирует правильное присвоение меток данным. Устройство с одной меткой обычно полагается на внешнее присвоение меток.

Примером многоуровневого устройства может служить жесткий диск. У всех данных, помещаемых на жесткий диск, есть метки секретности. Примером устройства с одной меткой может служить принтер, физически расположенный в среде, для входа в которую нужен допуск по защите. На принтер могут быть отправлены только те данные, у которых есть этот допуск.

Предупреждения относительно разработки драйверов устройств:

Драйверы устройств входят в состав ядра операционной системы и поэтому не ограничены в своих действиях. Создание или изменение драйверов устройств так же влияет на защиту, как и модификация самого ядра. К сожалению, пользователям часто требуется создавать или изменять драйверы устройств. Это следует делать очень осторожно.

Невозможно перечислить все конкретные меры предосторожности, которые требуется соблюдать при написании драйверов устройств, поскольку драйверы могут нарушить защиту системы множеством различных способов (иногда - совершенно непреднамеренно). Таким образом, создание безопасных драйверов устройств во многом зависит от искусства и опыта разработчиков.

Драйверы устройств не должны выполнять никаких операций, кроме простого управления устройствами. Драйверы устройств, созданные прежде всего для добавления новых системных вызовов в систему, включая многие драйверы псевдоустройств, таких как `/dev/kmem`, должны рассматриваться как новые системные вызовы и разрабатываться соответствующим образом. Рекомендации этого раздела относятся преимущественно к тем драйверам, которые являются стандартными администраторами устройств.

Прежде чем создавать новые драйверы устройств, вы должны изучить существующие стандартные драйверы. Основные действия по защите, осуществляемые драйверами устройств, - это действия, связанные с выполнением системных вызовов **open** и **ioctl**.

Открытие устройств:

Как и для большинства системных объектов, основная часть процедур проверки защиты, связанных с доступом к устройству, выполняется при открытии устройства системным вызовом **open**.

Ядро сначала выполняет набор основных операций, а затем передает обработку запроса на открытие драйверу устройства. Перед тем, как передать управление драйверу устройства, ядро выполняет следующие процедуры проверки защиты:

- Если у процесса нет доступа MAC к специальному файлу устройства, открытие не выполняется
- Если у процесса нет доступа MIC к специальному файлу устройства, открытие не выполняется
- Если у процесса нет доступа DAC к специальному файлу устройства, открытие не выполняется

На многих устройствах чтение из устройства (посредством системного вызова **read**) вызывает такое изменение его состояния, которое может быть обнаружено другим процессом, метка MAC которого не поглощает метку считывающего процесса. Это образует потенциальный скрытый канал. Это проблема устройств типа FIFO ("первый вошел - первый вышел"). В таких случаях, как правило, доступ на чтение разрешают только процессам того же уровня MAC, что и устройство. В этом случае драйвер устройства выполняет соответствующую проверку.

Существует также несколько конкретных правил и рекомендаций, относящихся к проектированию нестандартных устройств. Вы должны усвоить и применять основные принципы обязательного и избирательного контроля доступа. К счастью, большинство драйверов устройств можно настроить как стандартные, и с особенностями нестандартных драйверов приходится сталкиваться не так часто.

Примеры открытия драйверов устройств:

Ниже приведены примеры обработки нестандартных устройств, взятые из стандартных системных драйверов устройств. Они иллюстрируют разнообразие этих драйверов устройств.

/dev/null

`/dev/null` - это псевдоустройство, не содержащее данных. Данные, записываемые в `/dev/null`, аннулируются и в отчет на запросы на чтение всегда возвращается символ конца файла (EOF). Таким

образом, ограничение устройства MAC на открытие не требуется. Для обеспечения совместимости доступ DAC к файлу устройства `/dev/null` обязателен, хотя и не является абсолютно необходимым.

/dev/tty

Когда процесс выдает запрос на открытие для `/dev/tty`, драйвер устройства фактически пытается открыть управляющий терминал запрашивающего процесса. Таким образом, необходимо проверять доступ MIC, MAC и DAC для процесса управляющего терминала процесса вместо `/dev/tty`. Для обеспечения совместимости доступ к `/dev/tty` обязателен, хотя и не является абсолютно необходимым.

Ограничения ioctl:

Хотя все функции интерфейсов драйверов устройств должны быть надежными, интерфейс **ioctl** обычно требует особого внимания.

Общее правило таково: только процессы с правами на запись могут изменять характеристику файла, распознаваемую другими процессами, не имеющими прав на запись. Обладание правами на запись означает, что либо процесс открыл файл для записи, либо метка MAC процесса равна метке устройства. Это ограничение основывается на общем ограничении MAC, которое гласит, что никакой процесс не может выполнять действие, которое может быть обнаружено процессами с меньшими метками MAC.

Если цель действия - чтение или запись пользовательских данных, то это ограничение должно быть применено в указанной форме. Иначе, случаи, когда это ограничение не применяется, рассматриваются как скрытые каналы и должны быть ограничены по пропускной способности и/или контролируемы.

Некоторые действия по управлению устройствами может потребоваться ограничить уровнем привилегированных процессов, даже когда устройство не настроено как надежное.

Прочие ограничения:

Существует сравнительно немного других случаев, когда драйверу устройства может потребоваться применять специальные меры защиты.

Один из примеров - когда чтение из устройства вызывает такое изменение его состояния, которое может быть обнаружено другим процессом, метка MAC которого не поглощается меткой считывающего процесса. Это представляет собой потенциальный скрытый канал, который может потребоваться ограничить или контролировать самим драйвером устройства.

Обзор программирования драйверов устройств:

При реализации драйверов устройств учтите следующие рекомендации.

Примечание: Добавлены новые системные вызовы, поддерживающие расширенную защиту каждой операции чтения или записи в потоковых устройствах и устройствах FIFO. Два новых библиотечных API, eread() и ewrite(), поддерживают этот атрибут расширенной защиты. В случае ядра MLS на устройстве устанавливается флаг защиты DEV_SEC_ERDWR. Аналогично, на устройстве FIFO устанавливается флаг GNF_SEC_ERDWR. Эти флаги обеспечивают дополнительные меры защиты в каждой операции чтения или записи.

Общие приемы проектирования

Все меры защиты в драйвере устройства должны быть написаны в соответствии с принципом модульности и должны быть легко распознаваемы.

Меры защиты в драйверах устройств

Проверки MIC, MAC и DAC всегда лучше выполнять вне драйверов устройств. Драйверы устройств без таких мер защиты легко переносимы в ненадежные системы и надежные системы других типов и из таких систем.

В стандартной реализации драйвера устройства проверки MIC, MAC и DAC выполняются ядром, а все дополнительные обязательные проверки привилегий - драйвером. В нестандартной реализации драйвера устройства все проверки (MIC, MAC, DAC и проверки привилегий) выполняются драйвером. Выбор между стандартной и нестандартной реализациями драйвера устройства зависит от подхода к проектированию.

DAC

DAC применяется к каждому специальному файлу устройства в соответствии с точкой входа файловой системы, через которую осуществляется доступ к устройству.

Проверка правильности установки

Любой драйвер устройства, выполняющий проверки MAC, должен обеспечивать безопасность (в разумных пределах) в случае неправильного определения устройства.

Привилегированный доступ

Может возникнуть необходимость в том, чтобы драйвер устройства разрешал выполнение определенных операций над устройством только привилегированным процессам. Однако в этих ситуациях необходимо соблюдать несколько особых рекомендаций.

Определить, обладаете ли вы необходимыми привилегиями, можно с помощью функции ядра **refmon**.

Принцип наименьших привилегий:

В Trusted AIX вводится концепция наименьших привилегий. Принцип наименьших привилегий превращает ранее могущественного пользователя root в механизм предоставления привилегий с большей степенью детализации. Такое разбиение привилегий гарантирует, что в случае программной ошибки или другого дефекта в надежном программном обеспечении защите системы может быть нанесен лишь незначительный ущерб.

Операции над привилегиями:

С каждым процессом связано четыре вектора привилегий: действующий, максимальный, наследуемый и предельный.

Максимальный вектор привилегий определяет верхнюю границу привилегий, которые могут быть активны для каждого процесса. Действующий вектор привилегий - привилегии, которые изучаются для принятия решения о привилегиях. Учтите, что действующий набор привилегий всегда является подмножеством максимального набора привилегий, который, в свою очередь, является подмножеством предельного набора привилегий. Предельный набор определяет привилегии, которые могут входить в максимальный, наследуемый и действующий наборы. Наследуемый набор привилегий представляет набор привилегий, наследуемых дочерними процессами в операциях fork и exec.

При обработке нового текстового образа нарастание привилегий происходит по следующему алгоритму. Специальные привилегии - это **PV_ROOT**, **PV_SU_**, **PV_SU_EMUL**, **PV_SU_ROOT**, **PV_AZ_ROOT** и **PV_SU_UID**.

Следующий алгоритм демонстрирует две важные концепции подсистемы наименьших привилегий. Первая заключается в том, что специальные привилегии (**PV_ROOT**, **PV_SU_**, **PV_SU_EMUL**, **PV_SU_ROOT**,

PV_AZ_ROOT и **PV_SU_UID**) - это единственные привилегии, которые могут распространяться без дополнительных условий во время обработки нового образа процесса. Вторая - в том, что действующий вектор привилегий процесса освобождается от всех привилегий, если только в файле не задано **FSF_EPS**. Это гарантирует совместимость с предыдущими версиями приложений, которые при необходимости можно запускать в надежной системе, в которой применяется принцип наименьших привилегий.

```
new_max_privs = old_inheritable_privs
new_max_privs = new_max_privs | file_innate_privs
IF (пользователю были присвоены некоторые права доступа в файле PAS)
new_max_privs = new_max_privs | file_authorized_privs
new_max_privs = new_max_privs & old_limiting_privs
IF (old_max_privs содержит одну или несколько специальных привилегий)
new_max_privs += тот же набор специальных привилегий
IF (для исполняемого файла задано FSF_EPS)
new_eff_privs = new_max_privs
ELSE
new_eff_privs = old_inheritable_privs
IF (old_eff_privs содержит одну или несколько специальных привилегий)
new_eff_privs += тот же набор специальных привилегий
new_limiting_privs = old_limiting_privs
```

Присвоение и удаление прав доступа:

На примере описанных ниже стандартных процедур системной библиотеки проиллюстрирован процесс управления правами доступа в системе. Эти процедуры применимы только к привилегированным программам в системе.

priv_raise
: Изменение вектора эффективных прав доступа, присвоенных процессу, путем дополнения (или повышения) указанного списка прав доступа. Список прав доступа должен принадлежать к вектору максимальных прав доступа процесса, иначе будет выдано сообщение об ошибке.

priv_remove
: Изменение вектора эффективных или максимальных прав доступа, присвоенных процессу, путем удаления указанного списка прав доступа. Если процессу не удается удалить эффективные или максимальные права доступа, выдается сообщение об ошибке.

priv_lower
: Изменение вектора эффективных прав доступа, присвоенных процессу, путем удаления (или понижения) указанного списка прав доступа. Если процессу не удается понизить эффективные права доступа, выдается сообщение об ошибке.

Каждая из этих процедур воспринимает список прав доступа, разделенный запятыми, в конце которого ставится **-1** (минус один, недопустимый номер права доступа). Данный метод повышения и понижения прав доступа в рамках минимальной части кода, для которой могут потребоваться именно эти права доступа, называется заключением прав доступа в скобки. Прием заключения прав доступа в скобки следует применять во всех защищенных прикладных программах, чтобы снизить вероятность преодоления защиты из-за неправильно разработанного или реализованного программного обеспечения.

setppriv
: Изменение вектора эффективных, максимальных, наследуемых и ограничивающих прав доступа процесса, путем формирования наборов прав доступа. Если переданные наборы прав доступа недопустимы или не разрешены, выдается сообщение об ошибке.

Права доступа:

Права доступа предоставляют пользователям различные наборы привилегий.

Обычно команда или утилита проверяет имеющиеся права доступа в начале выполнения и затем в соответствии с этим задает свои собственные привилегии. Таким образом, пользователи с конкретными правами доступа получают свой набор привилегий для каждой выполняемой команды, в зависимости от того, как запрограммирована команда.

Для того чтобы громоздкие параметры привилегий не задавались в самом коде, AIX предоставляет наборы прав доступа и наборы привилегий в виде внешних двоичных файлов. Благодаря Privileged Authorization Set (PAS) и Authorized Privilege Set (APS) система, а не команда, задает привилегии на основе прав доступа.

checkauths
Сравнивает значения, переданные в списке прав доступа, с правами доступа для текущего процесса.

Дополнительная информация о проверке прав доступа приведена в разделе “Права доступа RBAC” на стр. 86.

Контроль:

Trusted AIX включает в себя набор команд для управления составлением контрольного журнала и информацией в нем. Программисту защищенной системы вряд ли понадобится изменять эти программы или вносить в них дополнения.

audit Управляет демоном контроля

auditbin
Управляет файлами контрольного журнала

auditselect
Объединяет и выбирает контрольные записи из файлов контрольного журнала

auditpr
Представляет отдельные события контроля в виде, пригодном для прочтения пользователем

События контроля, генерируемые защищенными программами, представляют собой основную сферу, представляющую интерес для программиста защищенной системы. Большинство защищенных программ в обязательном порядке передают сообщения в контрольный журнал системы.

Ситуации для контроля:

Для определения, какие ситуации следует выявлять и контролировать с помощью защищенной программы, точных инструкций почти нет. Главным образом, это вопрос рассуждений и стратегии контроля. Базовая операционная система делит ситуации на успешное выполнение, невыполнение, доступ к объектам и возможные скрытые каналы.

Выполненные операции:

Выполненные ситуации необходимо контролировать для создания хронологии основного использования.

Например, важно, чтобы программа распределения устройств между пользователями регистрировала, когда данный пользователь занимает и освобождает устройство. В этом случае программа будет иметь возможность отслеживать прохождение информации в системе и определять ответственного в том случае, если впоследствии выяснится, что устройством злоупотребили. С другой стороны, в некоторых доктринах контроля успешным операциям практически не уделяется внимания, поскольку о них известно от защищенных программных средств, что операции являются законными и выполненными надлежащим образом.

Сбои:

Операции контроля, выполнение которых закончилось сбоем, могут пригодиться для выявления пользователей, которые пытались получить доступ к закрытым услугам или данным. Частое возникновение таких сбоев может указывать на злонамеренные действия сотрудников (если они не отличаются особыми талантами в этом отношении).

Согласно базовой системе, сбои контроля можно разбить на пять категорий:

- Сбои прав доступа (попытка несанкционированного выполнения определенного действия процессом, которому не присвоены соответствующие права доступа)
- Сбои MAC (сбой действия, вызванный тем, что выполнение данного действия будет означать нарушение ограничений MAC)
- Сбои MIC (сбой действия, вызванный тем, что выполнение данного действия будет означать нарушение ограничений MIC)
- Сбои DAC (сбой действия, вызванный тем, что выполнение данного действия будет означать нарушение ограничений DAC)
- Прочие сбои (например, попытка войти в систему с помощью неверного пароля)

Доступ к объектам:

Контролировать доступ к объектам необходимо для того, чтобы выявить пользователей, обращающихся к данному объекту (например, к теневому файлу пароля).

Потенциальные скрытые каналы:

Контроль потенциальных скрытых каналов - важный вопрос, поскольку скрытые каналы можно использовать для передачи информации от одного процесса к другому по разным меткам MAC. Слово использовать не означает, что эти каналы были использованы для указанной цели; просто такое использование не исключено.

Каждая запись, внесенная системой контроля, содержит причину появления записи о контроле (успех, ошибка MAC, ошибка MIC, ошибка DAC, ошибка прав доступа, другая ошибка, доступ к объекту или потенциальный скрытый канал). Это касается контрольных записей, внесенных как самой системой, так и пользовательскими программами.

Полезно учитывать, доверенный ли данный пользователь (то есть, является ли он администратором), однако нельзя совершенно точно установить, какой пользователь требует более пристального контроля - доверенный или не доверенный. Например, администраторы считаются доверенными пользователями, требующими, соответственно, меньше контроля, но в то же время их действия могут иметь далеко идущие последствия, и в таком случае полезно регистрировать действия администратора без соответствующих полномочий. Постоянные пользователи могут нанести меньший урон и в этом смысле требуют меньшего контроля, но при этом они не являются в той же степени доверенными, отчего могут требовать большего контроля. Системные администраторы зачастую применяют более строгий контроль своих действий, который мог бы подтвердить их невиновность в случае взлома системы защиты.

Объектами контролирования могут быть следующие события:

- Успешные операции, в особенности, операции передачи информации или изменение параметров управления доступом
- Операции, которые не удалось выполнить по причинам, связанным с защитой
- Операции, выполненные администраторами, - успешные и нет
- Потенциальное использование скрытых каналов
- Операции доступа к конкретному объекту
- Действия, которые влияют на последующее содержание фактического контрольного журнала

Уровни контрольной информации:

Контрольная информация высшего уровня является более нужной, чем контрольная информация никого уровня. Защищенные программы обеспечивают высокоуровневый обзор операций и предоставляют контрольные сообщения высочайшего качества.

Запись, содержащая сведения только о том, что файл защиты был открыт администратором для записи, значительно менее полезна, чем запись о факте выполнения высокоуровневой операции над файлом (например, запись о том, что администратором была создана новая запись в файле, с указанием ключевой информации об этой новой записи). Настоятельно рекомендуется формирование контрольной информации на как можно более высоком уровне.

Лучше включить информацию об одном событии, чем о нескольких. Основной причиной разделения процедуры выполнения контроля на несколько событий связана с возможностью выборочного включения отдельных процедур контроля.

Классы и события контроля:

Каждая защищенная программа должна определить класс контроля, тип события контроля и причину, на которую она ссылается при выдаче контрольных сообщений посредством системного вызова **auditlog**.

Каждое событие контроля относится к одному из классов контроля. Разбивая события на классы, можно повысить эффективность работы с большим количеством событий. Определения классов контроля содержатся в файле `/etc/security/audit/config`.

Класс контроля используется для включения и выключения записи событий. Если очень важно активизировать два события по отдельности, они не должны относиться к одному и тому же классу контроля. В целом, практика разделения событий на классы зарекомендовала себя как очень удачная. Как правило, за любой защищенной программой или группой связанных защищенных программ закрепляется одно имя класса контроля (или, в редких случаях, несколько имен классов контроля), для использования этой программой или группой программ.

Действия системы, подлежащие контролю, называются событиями контроля. Их определения приведены в файле `/etc/security/audit/events`.

Скрытые каналы:

Считается, что защищенное программное обеспечение не задействовано в схемах с применением скрытых каналов. Кроме того, такое программное обеспечение должно быть разработано таким образом, чтобы сделать невозможным его использование незащищенными программами с целью использования скрытых каналов. В этом разделе будут описаны скрытые каналы и предоставлены рекомендации по выявлению и ограничению их использования.

Определение скрытых каналов:

Процессу с меткой А запрещено выполнять действие, распознаваемое другим процессом с меткой Б, кроме случаев, когда метка Б является доминирующей по отношению к метке А.

Это определение можно рассмотреть, изучив по отдельности две ситуации: прямые операции с данными и случайные операции. Прямые операции с данными предназначены для пользователей и служат непосредственным способом хранения или передачи пользовательских данных, например, чтения или записи файлов. Эти операции должны целиком и полностью подчиняться базовому ограничению MAC. Все остальные операции являются случайными. Использование случайной операции для передачи данных вопреки базовому ограничению MAC называется скрытым каналом.

Для использования скрытого канала необходимо наличие двух незащищенных процессов, которые называются отправитель (с меткой X) и получатель (с меткой Y). Допустим, метка MAC получателя не является доминирующей по отношению к метке отправителя (если бы она была таковой, то поток данных от отправителя к получателю представлял бы собой вполне разрешенное повышение уровня). Для того, чтобы пользоваться этим каналом, и отправитель, и получатель следуют определенным конвенциям в отношении использования согласованных ресурсов в целях передачи данных вопреки ограничениям MAC.

Единственным критерием, указывающим на использование скрытого канала, служит то, что метка получателя не доминирует над меткой отправителя, и оба процесса, отправитель и получатель, являются незащищенными. И отправитель, и получатель, как правило, - это процессы, используемые одним и тем же пользователем. Предполагается, что защищенная компьютерная база (TCB) по определению поддерживает базовое ограничение MAC и не имеет кодов, допускающих нарушение этого ограничения путем злонамеренного использования скрытых каналов. (На самом деле, привилегированным процессам доступно множество гораздо более эффективных способов, позволяющих нарушить ограничения MAC, даже не прибегая к использованию скрытых каналов.) Обеспокоенность вызывает именно способность незащищенных процессов пользоваться скрытыми каналами посредством защищенных программ.

Вообще, следует исключить наличие скрытых каналов в системе. Тем не менее, бывают ситуации, в которых потребности другой системы (такие как быстродействие, надежность или совместимость) сталкиваются с неприемлемыми ограничениями как раз из-за отсутствия скрытых каналов.

Рекомендации относительно пропускной способности:

Базовая операционная система использует описанные ниже рекомендации по ограничению использования скрытых каналов, исходя из их пропускной способности:

Больше 100 битов в секунду
: Существование таких каналов не допускается

От 0,1 до 100 битов в секунду
: Каналы в данном диапазоне могут существовать в случае абсолютной необходимости, но их использование подлежит выявлению и контролю всегда, если это возможно

Менее 0,1 бита в секунду
: Каналы данного диапазона могут существовать, если это необходимо, но выявлять их использование не обязательно

Настоятельно рекомендуется выполнение этих правил всеми дополнительными программами защищенной компьютерной базы (TCB). В дополнение, подумайте о том, что даже относительно медленные каналы с пропускной способностью в 10 битов в секунду могут передавать по 4500 байтов в час, а это значительный объем данных, уровень защиты которых несанкционированно понижается. Следовательно, нужно приложить все усилия к тому, чтобы максимально ограничить пропускную способность скрытых каналов.

Пропускная способность большинства скрытых каналов обычно снижается благодаря работе других процессов, а не только тех, которые могут незаконно использовать данный канал. Тем не менее, не стоит считать данный метод ограничения пропускной способности скрытых каналов достаточно надежным, поскольку в любой системе бывают периоды снижения активности.

Выявление скрытых каналов:

Выявление скрытых каналов в основном зависит от тщательного анализа и разработки. Ниже приводится несколько специальных рекомендаций по выявлению скрытых каналов.

Термин модуль означает единицу кода защищенной компьютерной базы (TCB), выявляющего или ограничивающего использование скрытого канала, в ядре или в процессе. Выявление скрытых каналов заключается в основном в определении того, может ли незащищенный процесс (отправитель) на уровне A использовать модуль для выполнения действия, распознаваемого другим процессом (получателем) на уровне Б, если уровень Б не является доминирующим над уровнем A.

Например, обычный скрытый канал представляют собой данные, записываемые в файл защищенным процессом от имени незащищенного пользователя, если метка MAC данного файла не является доминирующей над меткой MAC данного пользователя.

Разработано относительно немного методик выявления скрытых каналов. Наиболее знаменитой из них является Матрица общих ресурсов (SRM). Описание этой методики можно найти в следующих источниках:

- • 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.

Выявление скрытого канала посредством контроля:

Возможность контролировать потенциальное использование скрытого канала может служить эффективной мерой защиты от этой угрозы. Тем не менее, для успешного контроля необходимо, чтобы событие контроля происходило относительно нечасто. Контроль считается недостаточно эффективным, если для события, инициировавшего выполнение контроля, показатель отношения фактического несанкционированного использования к случайному обращению является низким.

Ограничение использования скрытых каналов:

Лучшим способом ограничения использования скрытых каналов является их устранение.

Иначе, их использование следует ограничивать, следуя инструкциям, изложенным в Рекомендациях относительно пропускной способности. Кроме того, всегда, если это возможно и эффективно, потенциальное использование каналов следует контролировать.

В целом, посредством кодов ядра или драйверов устройств, предназначенных для достижения высокой эффективности и имеющих высокую пропускную способность, вряд ли удастся ограничить использование скрытых каналов. С помощью защищенных процессов легче ограничивать использование скрытых каналов.

Примечание: Нет необходимости ограничивать использование скрытых каналов процессами, имеющими одинаковые метки, или если процесс- получатель доминирует над отправителем. Следовательно, большинство модулей TCB позволяет повысить быстродействие системы путем отмены ограничений для таких ситуаций.

Квоты по меткам:

Многие скрытые каналы предполагают использование пула ресурсов, общего для процессов с различными метками MAC. Эффективно противостоять этому можно путем создания отдельных пулов ресурсов фиксированного размера для каждой метки MAC, чтобы процесс мог регулировать использование ресурсов из пула только для своей метки MAC.

Неиспользуемые ресурсы могут быть постепенно перемещены из одного пула в другой для удобства при удовлетворении динамических потребностей. Такой перенос ресурсов сам по себе является скрытым каналом, но с гораздо меньшей пропускной способностью, легко поддающейся ограничению.

Запаздывание:

Для ограничения скрытых каналов служит способ, при котором TCB следит за тем, чтобы при выполнении службы, где существует канал, прошло определенное время. Для этого модуль должен просто находиться в ожидании в течение установленного времени, которое можно рассчитать на основе объема передаваемой информации.

Однако во многих случаях программы, использующие скрытые каналы, могут противодействовать запаздываниям, выполненным не надлежащим образом. Например, процессы, пользующиеся такими

каналами, могут создавать по нескольку набору процессов отправки/получения. Несмотря на то, что TCB легко может ограничить каждый набор до определенной пропускной способности, совокупная пропускная способность по всем наборам равняется пропускной способности одного этого канала.

Лучше, когда конкретная служба TCB следит за применением запаздываний (тем или иным образом) ко всем процессам, которые могут пользоваться службой.

Запаздывания полезны для ограничения, но они уязвимы для достаточно простых мер противодействия злоумышленных программ, и потому разрабатывать их нужно тщательно.

Ограничения данных:

Пропускную способность скрытого канала можно уменьшить не только путем увеличения времени, но и путем уменьшения объема возвращаемой информации. Программы, возвращающие данные в виде серии операций, в рамках одного промежутка времени часто могут возвращать меньшие пакеты информации или в меньшем количестве.

Приблизительное время:

Для многих приемов использования скрытых каналов необходимо, чтобы процессы, использующие эти каналы, включали в себя способ точного измерения относительного или абсолютного времени. Иногда удается ограничить возможности использования скрытых каналов, воспрепятствовав точному определению времени процессом.

Сделать так, чтобы службы защищенной компьютерной базы возвращали лишь приблизительную информацию о времени, относительно легко. Тем не менее, в распоряжении процессов иногда имеются другие способы измерения промежутков времени, например, путем оценки времени выполнения собственных команд. К подобным приемам ограничения использования скрытых каналов следует прибегать с осторожностью.

Источники шума:

Пропускная способность большинства скрытых каналов обычно снижена, иногда в значительной степени, из-за работы других процессов, а не только тех, которые используют данный канал. Можно, хотя и не рекомендуется, создавать защищенные программы, специально предназначенные для постоянного поддержания активности на определенном уровне. Иногда такие программы называются источниками шума.

Хотя идея использования источников шума может показаться привлекательной с концептуальной точки зрения, но обычно такие программы не способны определить, когда нужно создавать шум, а когда - нет. Следовательно, данный прием не рекомендуется применять для ограничения использования канала.

Цепочки U-T-U:

В некоторых ситуациях незащищенный процесс **U1** вызывает привилегированный, защищенный процесс **T**, который, в свою очередь, вызывает другой незащищенный процесс **U2** с другой меткой, чем **U1**. **U1** и **U2** представляют незащищенные процессы по разным меткам MAC, которые особенно способствуют наличию скрытого канала ввиду того, что один процесс является потомком другого. (Фактически, T и U могут быть последовательностями защищенных и/или незащищенных процессов.) Такую ситуацию называют цепочкой U-T-U.

Защищенные процессы должны следить за тем, чтобы информация не проходила между двумя незащищенными процессами в соответствии с основным принципом MAC, который предусматривает исключение запрещенных прямых операций с данными и скрытых каналов. Следует принимать во внимание:

- Дескрипторы файлов нельзя оставлять открытыми, когда **U2** не удалось открыть файл в режиме чтения/записи, в котором он открыт

- Переменные среды нужно очистить, если метка **U2** не доминирует над U1
- Рабочий каталог, переданный из **U1** в **U2**, может служить скрытым каналом (вероятно, небольшим), если метка **U2** не доминирует над **U1**. Аналогичным образом, многие параметры процессов, автоматически унаследованные дочерним процессом, могут служить скрытым каналом.

Цепочками U-T-U можно управлять надлежащим образом (то есть, скрытые каналы можно в значительной мере ограничить). Однако его трудно обеспечить, поэтому лучше избегать цепочек U-T-U. При этом обратите внимание: все дело в том, что незащищенным является U2--он без риска может быть защищенным, но при этом не привилегированным.

Примеры скрытых каналов:

Ниже приводятся примеры скрытых каналов, существование которых возможно в модулях, созданных системным программистом.

Пример скрытого канала службы печати:

Приведен пример скрытого канала службы печати.

Служба печати доверенной линии правильно помечает каждое переданное на выполнение задание меткой MAC запрашивающего процесса и сохраняет эту метку для заданий очереди, которая будет использована позднее при печати. Можно использовать относительно длинные имена заданий.

Программа состояния позволяет пользователю видеть все свои задания, которые находятся в очереди, вместе с именем задания, присвоенным пользователем, вне зависимости от метки задания. Это может служить скрытым каналом, поскольку в таком случае процесс отправителя может создавать задания, в имени которых содержатся данные для скрытой пересылки получателям, которые работают под именем того же пользователя.

Примечание: Единственным критерием использования скрытого канала является то, что метка отправителя не доминирует над меткой получателя, и что отправитель, так же, как и получатель, не являются доверенными пользователями. Отправитель и получатель чаще всего работают под именем одного и того же пользователя.

Такой канал можно закрыть, если предоставить пользователю возможность просматривать только те задания, над которыми доминирует метка MAC текущего пользователя. Тогда метка MAC получателя будет доминировать над соответствующей меткой отправителя, и канал можно будет использовать только для разрешенной модернизации. В порядке любезности программа состояния могла бы выдавать пользователю сообщение "есть другие задания" при наличии заданий, над которыми не доминирует метка. Получится гораздо более узкий канал, имеющий основания для существования.

Примечание: Иногда полезно контролировать выявление заданий более высокого уровня, так как при нормальной работе такое выявление будет выполняться, вероятно, в редких случаях.

Это распространенный пример скрытого канала, когда к объектам данных с многоуровневыми именами (в данном случае - задания на печать из очереди) процессы могут обращаться по разным меткам MAC. Канал можно эффективно закрыть, если применить метку MAC объекта также и к имени. Скрытую информацию могут нести также другие атрибуты помимо имени, например, размер.

Пример пула ресурсов:

Когда защищенная программа выполняет служебную функцию для незащищенного клиента, она выделяет особый тип ресурсов (например, буфер) из пула ресурсов, общего для процессов с разными метками MAC.

Здесь можно организовать скрытый канал, например, если для отправителя и получателя будут выделены все ресурсы, кроме одного, возможно, другими программами, работающими с другой или разными метками

MAC или другим или разными ИД пользователя. В этом случае отправитель делает последний оставшийся ресурс выделенным или невыделенным, а получатель обнаруживает это, также пытаясь сделать ресурс выделенным.

Это классический пример канала с использованием общих ресурсов. Пресечь такой канал можно за счет использования пулов ресурсов, помеченных метками, как указывалось выше. Обнаружить такой канал можно также с помощью контроля.

Пример баз данных:

Защищенная система базы данных позволяет помещать данные из пользовательских программ в многоуровневую базу данных. Контроль над прямым доступом надлежащим образом осуществляется посредством основных ограничений MAC.

Тем не менее, время, необходимое для помещения записи в базу данных, в значительной степени зависит от текущего общего размера базы данных. Следовательно, отправитель может разместить или удалить записи, что повлияет на размер базы данных, а получатель может лишь определить этот размер, оценив время, затрачиваемое на размещение записи. При недостаточно рациональной организации доступа к базе данных пропускная способность такого канала, скорее всего, будет низкой.

Для установления ограничений на использование канала можно задать гарантированное минимальное время доступа. Время задержки может быть псевдослучайным, чтобы сократить средние потери времени. Тем не менее, это всего лишь схема, основанная на времени задержки, и реализовывать ее следует с осторожностью.

Простой контроль над всеми видами доступа вряд ли может быть эффективным, поскольку очень сложно выявить несанкционированное использование канала, если одновременно с базой данных работает множество добросовестных пользователей.

Примеры программирования:

В этом разделе приведено несколько примеров программирования защищенных программ

Пример проверки прав доступа к защищенной программе:

Представлена модульная процедура для защищенной программы, которая позволяет проверить, имеет ли вызывающий процесс конкретные права доступа.

```
#include <sys/priv.h>
#include <sys/secattr.h>

int
priv_check (int priv)
{
  /* атрибуты защиты процесса */
  secattr_t secattr;

/* получение атрибутов защиты вызывающего процесса */
  if ( sec_getpsec(-1, &secattr;) != 0 )
  {
    return (-1);
/* ошибка получения структуры разрешений процесса */
  }

  /*
   * возврат данных о том, имеются ли указанные права доступа в
   * максимальном наборе прав доступа вызывающего процесса
   */
  return privbit_test(secattr.sc_maxpriv, priv);
}
```

Пример изменения эффективной Метки чувствительности (SL):

Эта программа позволяет изменить эффективную метку чувствительности (SL) текущего процесса до наивысшего уровня полномочий в системе.

В исходном наборе прав доступа программы обязательными являются следующие:

- **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; /* Высший уровень SL */
 secattr_t attr;
 char *clBuffer = NULL;

 /*
  * Получить высшие и низшие SL системы.
  */
 if ((sec_getsyslab(NULL, &sl_syshi, NULL, NULL)) != 0 ) {
  fprintf (stderr, "Сбой вызова sec_getsyslab.\n");
  exit(ERROR);
 }

 /*
  * Инициализировать этот процесс с initlabeldb() для доступа к
  * системной базе данных меток по умолчанию.
  */
 priv_raise(PV_LAB_LEF,-1);
 if (initlabeldb(NULL) != 0) {
  fprintf(stderr,  "Невозможно чтение базы данных кодирования меток.\n");
  exit(ERROR);
 }
 priv_remove(PV_LAB_LEF, -1);

 /*
  * Получить диапазон допуска процесса и эффективную SL.
  */
 priv_raise(PV_LAB_SLUG, PV_LAB_SL_SELF, -1);
 if (sec_getpsec(-1, &attr) != 0) {
  fprintf (stderr, "При получении атрибутов защиты программы Защищенного AIX возникла неполадка.\n");
  exit(ERROR);
 }

 /* malloc для максимальной длины метки SL, которая может быть сформирована для процесса */
 if((clBuffer = (char *) malloc(maxlen_cl())) == NULL) {
  perror("malloc");
  exit(ERROR);
 }
 /* Преобразовать двоичный эффективную SL в формат, пригодный для чтения пользователем */
 if (clbtohr(clBuffer, &attr.sc_sl, HR_LONG) != 0) {
  fprintf(stderr, "Невозможно преобразовать SL в формат, пригодный для чтения пользователем.\n");
  exit(ERROR);
```

```
 }
 printf("Исходная эффективная SL программы = %s.\n",clBuffer);

 /*
  * Предоставить эффективной метке чувствительности (SL) процесса наивысший уровень полномочий в системе.
  * Возможно, процессу не присвоена максимальная SL с наивысшим уровнем полномочий в системе,
  * поэтому необходимо задать этот уровень.
  */
 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, "При установке эффективной SL программы возникла неполадка.\n");
  exit(ERROR);
 }

 priv_lower(PV_LAB_SLUG, PV_LAB_SL_SELF, -1);

 if (sec_getpsec(-1, &attr) != 0) {
  fprintf (stderr, "При получении атрибутов защиты программы Защищенного AIX возникла неполадка.\n");
  exit(ERROR);
 }

 /* Преобразовать двоичный эффективную SL в формат, пригодный для чтения пользователем */
 if (clbtohr(clBuffer, &attr.sc_sl, HR_LONG) != 0) {
  fprintf(stderr, "Невозможно преобразовать SL в формат, пригодный для чтения пользователем.\n");
  exit(ERROR);
 }
 printf("Измененная эффективная SL программы = %s.\n",clBuffer);
 return(SUCCESS);
}
```

Примеры задания классификаций меток чувствительности и сравнения меток чувствительности:

Приведен пример задания классификаций меток чувствительности и использования библиотечных процедур для сравнения меток.

Для набора прав доступа proxy программы и максимального набора прав доступа вызывающего процесса обязательным правом доступа является **PV_LAB_LEF**.

```
 #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)
 {
 /* Метки чувствительности */
 sl_t sl1, sl2;

 /* строки, в которых содержатся метки чувствительности */
 char *slBuffer1 = NULL;
 char *slBuffer2 = NULL;

 if (argc != 3) {
 fprintf(stderr, "Usage: compare slabel1 slabel2\n");
 exit(ERROR);
 }
 /*
 * Инициализировать этот процесс с initlabeldb() для доступа к
 * стандартной системной базе данных Метка.
 */
```

```
priv_raise(PV_LAB_LEF , -1);
if (initlabeldb(NULL) != 0) {
fprintf(stderr, "Could not read the Label Encodings Database.\n");
exit(ERROR);
}
priv_remove(PV_LAB_LEF, -1);

/* преобразование переданной SL в двоичный формат */
if (slhrtob(&sl1, argv[1]) != 0) {
fprintf(stderr, "Unable to convert %s to binary form.\n", argv[1]);
exit(ERROR);
}
if (slhrtob(&sl2, argv[2]) != 0) {
fprintf(stderr, "Unable to convert %s to binary form.\n", argv[2]);
exit(ERROR);
}

/* функция malloc для максимальной допустимой длины метки  SL */
slBuffer1 = (char *) malloc(maxlen_sl());
slBuffer2 = (char *) malloc(maxlen_sl());

if ((slBuffer1 == NULL) || (slBuffer2 == NULL)) {
perror("malloc");
exit(ERROR);
}

/*
* Обратное преобразование метки в форму пользовательского уровня (длинную форму).
* Необязательный шаг. Приводится в качестве примера
* использовния API slbtohr().
*/
if (slbtohr(slBuffer1, &sl1, HR_LONG) != 0) {
fprintf(stderr, "Unable to convert to binary human readable form.\n");
exit(ERROR);
}

if (slbtohr(slBuffer2, &sl2, HR_LONG) != 0) {
fprintf(stderr,"Unable to convert to binary human readable form.\n");
exit(ERROR);
}

/*
* Использование sl_cmp() для сравнения доминантности двух меток.
*/
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("The two labels are disjoint.\n");
}

return (SUCCESS);
}
```

Пример задания информации для контроля:

Данная программа получает и задает информацию для контроля.

В собственном наборе прав доступа программы должны быть следующие права:

- **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;
 /* *Получение маски предварительного выбора для контроля процесса */
 priv_raise(PV_AU_ADMIN, -1);
 rc = auditproc(0, AUDIT_QEVENTS, buf, sizeof (buf));
 priv_lower(PV_AU_ADMIN, -1);
 if (rc)
  fprintf(stderr, "Failed to get audit info\n");
        /* *Добавление класса контроля ядра к маске предварительного выбора */
 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, "Failed to set audit info\n");
 /* *Задание GID процесса для создания контрольной записи */
 priv_raise(PV_DAC_GID, -1);
 rc = setgid(129);
 priv_lower(PV_DAC_GID, -1);
 if (rc)
  fprintf(stderr, "Failed to setgid\n");
 exit(0);
}
```

Пример клиента:

Эта программа направляет два сообщения серверу, одно с использованием стандартной функции **write**, а второе - с использованием функции **ewrite**.

Защищенное сообщение направляется как SECRET. Обратите внимание на то, что незащищенное сообщение, направленное посредством вызова **write**, по умолчанию получает набор атрибутов защиты, который можно настроить с помощью команды netrule.

В исходном наборе прав доступа программы обязательными являются следующие:

- **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,  "Формат:%s:  ADDR  PORT\n",  argv[0]);

  exit(1);
 }
#ifdef  SECURE
 /*
  * *  Получить доступ к базе данных кодирования меток (Label  Encodings  Database)
  *
  *     */


 priv_raise(PV_LAB_LEF,-1);
 l_init_result  =  initlabeldb(NULL);
 if  (  priv_remove(PV_LAB_LEF,  -1)  !=  0  )
 {
  fprintf(stderr,  "Сбой прав доступа\n");
  exit(1);
 }
 if  (  l_init_result  !=  0  )
 {
  fprintf(stderr,  "Невозможно чтение базы данных кодирования меток\n");
  exit(0);
 }
#endif  /*SECURE*/
  /*
   * *  Заполните структуру "serv_addr"  адресом
   *
   * *  сервера, к которому необходимо подключиться.
   * */
 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]));
 /*  Открыть сокет TCP   (сокет потока Internet).  */
 if  (  (sockfd  =  socket(AF_INET,  SOCK_STREAM,  0))  <  0)
 {
```

```
 perror("tcpclient:  ");
 fprintf(stderr,  "client:  Не удается открыть сокет потока\n");
 exit(0);
}
if  (  connect(sockfd,  (struct  sockaddr  *)  &serv_addr;,
   sizeof(serv_addr))  <  0  )
{
 perror("tcpclient:  ");
 fprintf(stderr,  "client:  Не удается подключиться к серверу\n");
 exit(0);
}
/*
 * *  Отправить на сервер обычное сообщение write,  которому
 * *  будут присвоены атрибуты защиты по умолчанию
 * */
strcpy(buf,  "Этому сообщению присвоены атрибуты защиты по умолчанию.\n");
if  (  write(sockfd,  buf,  strlen(buf)+1)  ==  -1  )
{
 perror("tcpclient:  ");
 fprintf(stderr,  "write  error\n");
}
#ifdef  SECURE
 strcpy(buf,  "Это сообщение SECRET\n");
/*  Настроить SL и 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;
/*  Для этого вызова  ewrite  необходимы права доступа  PV_MAC_CL  и 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,  "ewrite  error\n");
}
fflush(stderr);
#endif  /*SECURE*/
fprintf(stderr,  "exiting  .....  \n");
sleep(3);
close(sockfd);
exit(0);
}
```

Пример сервера:

Данная программа выступает в роли сервера и получает сообщения, отпаравленные на ее порт, с помощью процедуры **eread**. После успешного получения сообщения программа выводит атрибуты защиты сообщения.

В собственном наборе прав доступа программы (без расстановки специальных флагов FSF_EPS) должны быть следующие права:

- **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  /*  ЗАЩИЩЕННЫЙ  */
 if  (  argc  !=  2  )
 {
  fprintf(stderr,  "Usage:%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,  "Privilege  Failure\n");
  exit(1);
 }

 if  (l_init_result  !=  0)
 {
  fprintf(stderr,  "Could  not  read  the  Label  Encodings  Database\n");
  exit(1);
 }
#endif  /*  ЗАЩИЩЕННЫЙ  */
 /*  Открывание сокета  TCP  (сокет  Internet-потока).  */
 if  (  (sockfd  =  socket(AF_INET,  SOCK_STREAM,  0))  <  0  )
 {
  perror("tcpserver:  ");
  fprintf(stderr,  "server:  Cant  open  stream  socket\n");
  exit(1);
 }
 /*Связывание локального адреса, чтобы клиент мог отправлять нам сообщения*/
 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:  Cant  bind  local  address\n");
  exit(0);
 }
 listen(sockfd,  5);
 for  (;;)
 {
  /*
   * *  Ожидание соединения с процессом клиента.
   * */
  fprintf(stdout,  "Waiting  for  a  connection  from  a  client\n");
```

```
  clilen  =  sizeof(cli_addr);
  newsockfd  =  eaccept(sockfd,  (struct  sockaddr  *)  &      cli_addr,
    &clilen;,  &seclab;);
  if  (  newsockfd  <  0  )
  {
   perror("tcpserver:  ");
   fprintf(stderr,  "server:  accept  error\n");
  }
  /*  Print  SL  */
  if  (  slbtohr(label_str,  &seclab.sl;,  HR_SHORT)  !=  0  )
  {
   fprintf(stderr,"problem  converting  sl  to  string\n");
  }
  else
  {
   fprintf(stdout,  "sl  =  %s.\n",label_str);
  }
  /*  Печать  МИН ЗАЗОРА  */
  if  (  slbtohr(label_str,  &seclab.sl_cl_min;,  HR_SHORT)  !=  0  )
  {
   fprintf(stderr,"problem  converting  min  clearance  to  string\n");
  }
  else
  {
   fprintf(stdout,  "sl_cl_min  =  %s.\n",label_str);
  }

  /*  Печать  МАКС ЗАЗОРА  */
  if  (  slbtohr(label_str,  &seclab.sl_cl_max;,  HR_SHORT)  !=  0  )
  {
   fprintf(stderr,"problem  converting  max  clearance  to  string\n");
  }
  else
  {
   fprintf(stdout,  "sl_cl_max  =  %s.\n",label_str);
  }
  if  (  (childpid  =  fork())  <  0  )
  {
   perror("tcpserver:  ");
   fprintf(stderr,  "server:  fork  error\n");
   exit(0);
  }
  else  if  (  childpid  ==  0  )  /*  дочерний процесс  */
  {
   int  i,  j;
   char  buf[BUFSIZ];
#ifdef  SECURE
  sec_labels_t  e_seclab;
#endif  /*  ЗАЩИЩЕННЫЙ  */
  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");
      /*  Print  SL  */
      if  (  slbtohr(label_str,  &e_seclab.sl;,  HR_SHORT)  !=  0  )
      {
   fprintf(stderr,  "problem  converting  sl  to  string\n");
       }
      else
      {
   fprintf(stdout,  "sl  =  %s.\n",label_str);
       }
       /*  Печать  МИН ЗАЗОРА  */
       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);
       }
       /*  Печать  МАКС ЗАЗОРА  */
      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  /*  НЕЗАЩИЩЕННЫЙ  */
        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  /*  НЕЗАЩИЩЕННЫЙ  */
              }
 }
 /*  родительский процесс  */
 close(newsockfd);
        }
}
```

Атрибуты защиты для пользователей и портов защищенной AIX:

Атрибуты защиты для пользователей и портов используются для получения атрибутов зазоров для пользователей и портов и сравнения атрибутов зазоров для пользователей с атрибутами для портов.

В файле **usersec.h** для Trusted AIX определены следующие дополнительные атрибуты.

S_MINSL
Метка минимального зазора чувствительности для пользователя. Введите SEC_CHAR

S_MAXSL
Метка максимального зазора чувствительности для пользователя. Введите SEC_CHAR

S_DEFSL
Метка стандартной чувствительности для пользователя. Введите SEC_CHAR

S_MINTL
Метка минимального зазора целостности для пользователя. Введите SEC_CHAR.

S_MAXTL
Метка максимального зазора целостности для пользователя. Введите SEC_CHAR.

S_DEFTL
Метка стандартной целостности для пользователя. Введите SEC_CHAR

Для портов допускаются следующие атрибуты.

S_MINSL
Метка минимальной чувствительности для порта. Введите SEC_CHAR.

S_MAXSL
Метка максимальной чувствительности для порта. Введите SEC_CHAR

S_TL Метка целостности для порта. Введите SEC_CHAR

В примере ниже определяется, может ли пользователь зарегистрироваться на конкретном порту.

```
#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, "Usage: %s <username> <portname>\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,
    "Error retriving attributes for user %s\n", username);
  exit(1);
 }

 if (clhrtob (&(userlab->minsl), attributes[0].attr_char)) {
  fprintf(stderr, "minsl conversion error\n");
  exit(1);
 }

 if (clhrtob(&(userlab->maxsl), attributes[1].attr_char)) {
  fprintf(stderr, "maxsl conversion error\n");
  exit(1);
 }

 if (clhrtob(&(userlab->defsl), attributes[2].attr_char)) {
  fprintf(stderr, "defsl conversion error\n");
  exit(1);
 }

 if (tlhrtob(&(userlab->mintl), attributes[3].attr_char)) {
  fprintf(stderr, "mintl conversion error\n");
  exit(1);
 }

 if (tlhrtob(&(userlab->maxtl), attributes[4].attr_char)) {
  fprintf(stderr, "maxtl conversion error\n");
  exit(1);
 }

 if (tlhrtob(&(userlab->deftl), attributes[5].attr_char)) {
  fprintf(stderr, "deftl conversion error\n");
  exit(1);
```

```
 }

 printf("User %s has the following clearance values\n", username);
 printf("minsl:%s\n", attributes[0].attr_char);
 printf("maxsl:%s\n", attributes[1].attr_char);
 printf("defsl:%s\n", attributes[2].attr_char);
 printf("mintl:%s\n", attributes[3].attr_char);
 printf("maxtl:%s\n", attributes[4].attr_char);
 printf("deftl:%s\n", attributes[5].attr_char);

 return;
}

void getportlabels(char *portname, struct portlabels *portlab)
{
 int rc =0;
 char *val = NULL;
 if ( (rc = getportattr(portname,S_MINSL,(char*)&val;, SEC_CHAR)) != 0 ) {
  perror ("Error retrieving port attributes");
  exit(1);
 }

 if (slhrtob(&(portlab->minsl), val)) {
  fprintf(stderr, "port minsl conversion error\n");
  exit(1);
 }

 if ( (rc = getportattr(portname,S_MAXSL, (char*)&val;, SEC_CHAR)) != 0 ) {
  perror ("Error retrieving port attributes");
  exit(1);
 }

 if (slhrtob(&(portlab->maxsl), val)) {
  fprintf(stderr, "port maxsl conversion error\n");
  exit(1);
 }

 if ( (rc = getportattr(portname,S_TL, (char*)&val;, SEC_CHAR)) != 0 ) {
  perror ("Error retrieving port attributes");
 }

 if (tlhrtob(&(portlab->tl), val)) {
  fprintf(stderr, "port tl conversion error\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("Default SL of user does not dominate the minimum SL of tty \n");
  exit(1);
 }

 cmpres = sl_cmp(&(portlab->maxsl), &(usrlab->defsl));
 if (cmpres != LAB_DOM && cmpres != LAB_SAME) {
  printf("Default SL of user is not dominated by maximum SL of tty \n");
  exit(1);
 }

 cmpres = tl_cmp(&(portlab->tl), &(usrlab->deftl));
 if (cmpres != LAB_SAME) {
  printf("Default TL of user is not same as TL of tty \n");
```

```
   exit(1);
  }

  printf("The user can login on the specified port\n");
  return;

}
```

Системные вызовы защищенной AIX:

Системные вызовы служат для работы с дополнительными функциями Trusted AIX.

eaccept
Принимает соединение в сокете

ebind Расширение связываний для работы с атрибутами защиты

econnect
Инициирует соединение на сокете, расширенном для работы с атрибутами защиты

eread Считывает данные из потока и получает атрибуты защиты сообщений

ereadv Считывает данные из потока и получает атрибуты защиты сообщений

erecv Расширение recv, recvfrom, recvmsg для работы с атрибутами защиты

erecvfrom
Расширение recv, recvfrom, recvmsg для работы с атрибутами защиты

erecvmsg
Расширение recv, recvfrom, recvmsg для работы с атрибутами защиты

esend Расширение send, sendto, sendmsg для работы с атрибутами защиты

esendmsg
Расширение send, sendto, sendmsg для работы с атрибутами защиты

esendto
Расширение send, sendto, sendmsg для работы с атрибутами защиты

ewrite Записывает данные в поток и задает атрибуты защиты сообщений

ewritev
Записывает данные в поток и задает атрибуты защиты сообщений

sec_getmsgsec
Получает атрибуты защиты очередей сообщений

sec_getpsec
Получает информацию защиты, связанную с процессом

sec_getrunmode
Получает режим работы ядра

sec_getsecconf
Возвращает флаги текущей конфигурации защиты

sec_getsemsec
Получает атрибуты защиты семафоров

sec_getshmsec
Получает атрибуты защиты сегментов общей памяти

sec_getsyslab
Получает стандартные метки чувствительности системы

sec_gettlibbufsize
Получает записи путей к библиотекам ядра

sec_gettlibpath
Получает записи путей к библиотекам ядра

pdmkdir
Создает/устанавливает/сбрасывает секционированный каталог или подкаталог

sec_setauditrange
Задает диапазон метки глобального контроля системы

sec_setplab
Устанавливает метку эффективной чувствительности, минимального зазора чувствительности, максимального зазора чувствительности, и метку целостности данных для указанного процесса

setppdmode
Задает режим секционированного каталога (реальный или виртуальный) для процесса

setppriv
Задает наборы прав доступаа, связанные с процессом

sec_setptlibmode
Задает режим TLIB для процесса

sec_setrunmode
Задает режим работы ядра

sec_setsecconf
Устанавливает флаги конфигурации защиты ядра

sec_setsemlab
Задает атрибуты защиты семафоров

sec_setshmlab
Задает атрибуты защиты сегментов общей памяти

sec_setsyslab
Устанавливает стандартные метки чувствительности системы, информации и целостности

Функции библиотеки AIX C:

Для управления дополнительными функциями Trusted AIX предусмотрены специальные подпрограммы и макрокоманды.

accredrange
Определить, находится ли метка чувствительности в пределах диапазона сертификации.

clbtohr
Преобразовать данную двоичную метку допуска в формат, пригодный для чтения пользователем

clhrtob
Преобразовать данную метку допуска из формата, пригодного для чтения пользователем, в двоичный формат

getfsfbitindex, getfsfbitstring
Подпрограммы для получения строк и индексов флага защиты файла

getmax_sl, getmax_tl
Получить метки максимальной чувствительности и целостности из файла кодировки меток.

getmin_sl, getmin_tl
Получить метки минимальной чувствительности и целостности из файла кодировки меток.

getsecconfig, setsecconfig
Подпрограммы для получения и установки флагов конфигурации защиты ядра для рабочих режимов

initlabeldb, endlabeldb
Подпрограммы запуска и прекращения работы базы данных меток.

maxlen_sl, maxlen_cl, maxlen_tl
Узнать максимальную длину меток в формате, пригодном для чтения пользователем, на основании инициализированного файла кодировки меток.

priv_isnull
Определить, имеются ли заданные права доступа в данном наборе прав доступа

priv_lower
Операции установки прав доступа

priv_raise
Операции установки прав доступа

priv_remove
Операции установки прав доступа

priv_subset
Операции установки прав доступа

privbit_clr
Удаление указанного права доступа из указанного набора прав доступа

priv_clrall
Удаление всех прав доступа из указанного набора прав доступа

priv_comb
Объединение первых двух указанных наборов прав доступа и помещение полученного набора в третий указанный набор прав доступа

priv_copy
Копирование первого указанного набора прав доступа во второй указанный набор прав доступа

priv_isnull
Определить отсутствие прав доступа в данном наборе прав доступа

priv_mask
Выявление точки пересечения между первыми двумя указанными наборами прав доступа и помещение полученного набора в третий указанный набор прав доступа

priv_rem
Удаление прав доступа, включенных во второй указанный набор прав доступа, из первого указанного набора прав доступа и помещение полученного набора в третий указанный набор прав доступа

privbit_set
Помещение указанного права доступа в указанный набор прав доступа

priv_setall
Помещение всех права доступа в указанный набор прав доступа

priv_subset
Определение того, является ли первый указанный набор прав доступа набором, вложенным во второй указанный набор прав доступа

privbit_test
Проверка с целью определения того, включено ли указанное право доступа в определенный набор прав доступа

slbtohr, clbtohr, tlbtohr
Подпрограммы преобразования меток из двоичного формата в формат, пригодный для чтения пользователем.

slhrtob, clhrtob, tlhrtob
Подпрограммы преобразования меток из формата, пригодного для чтения пользователем, в двоичный формат

sl_clr, tl_clr
Подпрограммы для сброса меток

sl_cmp, tl_cmp
Подпрограммы сравнения меток

tl_cmp Сравнение меток целостности

Права доступа в Trusted AIX

В Trusted AIX предусмотрены разные уровни прав доступа. В этом разделе приведено краткое описание различных прав доступа и их применения. Некоторые права доступа организованы в виде структур, когда предоставление какого-либо уровня доступа предусматривает также предоставление прочих связанных прав доступа.

При проверке прав доступа система всегда проверяет наличие минимальных необходимых прав доступа, а затем проверяет наличие прав доступа более высокого уровня. Например, процесс с правами доступа **PV_AU_** автоматически имеет также уровни доступа **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** и **PV_AU_WRITE**, а процесс с правами доступа **PV_ROOT** автоматически получает все перечисленные ниже права за исключением **PV_SU_**.

Привилегии контроля:

В Trusted AIX существуют следующие привилегии контроля. Ниже приводится краткое описание каждой привилегии и ее применения. Некоторые привилегии образуют иерархию, в которой одна привилегия может предоставлять все права, связанные с другой привилегией.

При выяснении привилегий система сначала проверяет, есть ли у процесса наименьшая необходимая привилегия, а затем движется вверх по иерархии, проверяя наличие более широких привилегий. Например, процесс с привилегией **PV_AU_** автоматически получает привилегии **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** и **PV_AU_WRITE**, а процесс с привилегией **PV_ROOT** - все перечисленные ниже привилегии, кроме привилегий **PV_SU_**.

PV_AU_
Эквивалентна объединению всех остальных привилегий **PV_AU_**

PV_AU_ADD
Позволяет процессу создавать и добавлять запись контроля

PV_AU_ADMIN
Позволяет процессу настраивать и запрашивать систему контроля

PV_AU_PROC
Позволяет процессу получить и задать состояние контроля процесса

PV_AU_READ
Позволяет процессу прочесть файл, помеченный как файл контроля

PV_AU_WRITE
Позволяет процессу записать или удалить файл, помеченный как файл контроля, или пометить файл как файл контроля

Привилегии прав доступа:

В Trusted AIX существуют следующие привилегии прав доступа. Ниже приводится краткое описание каждой привилегии и ее применения. Некоторые привилегии образуют иерархию, в которой одна привилегия может предоставлять все права, связанные с другой привилегией.

При выяснении привилегий система сначала проверяет, есть ли у процесса наименьшая необходимая привилегия, а затем движется вверх по иерархии, проверяя наличие более широких привилегий. Например, процесс с привилегией **PV_AU_** автоматически получает привилегии **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** и **PV_AU_WRITE**, а процесс с привилегией **PV_ROOT** - все перечисленные ниже привилегии, кроме привилегий **PV_SU_**.

PV_AZ_ADMIN
Позволяет процессу изменять таблицы защиты ядра

PV_AZ_READ
Позволяет процессу извлекать таблицы защиты ядра

PV_AZ_ROOT
Приказывает процессу передать проверки прав доступа во время системного вызова **exec**

PV_AZ_CHECK
Позволяет процессу передать все проверки прав доступа

Привилегии DAC:

В Trusted AIX существуют следующие привилегии DAC. Ниже приводится краткое описание каждой привилегии и ее применения. Некоторые привилегии образуют иерархию, в которой одна привилегия может предоставлять все права, связанные с другой привилегией.

При выяснении привилегий система сначала проверяет, есть ли у процесса наименьшая необходимая привилегия, а затем движется вверх по иерархии, проверяя наличие более широких привилегий. Например, процесс с привилегией **PV_AU_** автоматически получает привилегии **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** и **PV_AU_WRITE**, а процесс с привилегией **PV_ROOT** - все перечисленные ниже привилегии, кроме привилегий **PV_SU_**.

PV_DAC_
Эквивалентна объединению всех остальных привилегий PV_DAC_

PV_DAC_O
Позволяет процессу переопределять ограничения на принадлежность DAC

PV_DAC_R
Позволяет процессу переопределять ограничения на чтение DAC

PV_DAC_W
Позволяет процессу переопределять ограничения на запись DAC

PV_DAC_X
Позволяет процессу переопределять ограничение на выполнение DAC

PV_DAC_UID
Позволяет процессу задать или изменить свой ИД пользователя (UID)

PV_DAC_GID
Позволяет процессу задать или изменить свой ИД группы (GID)

PV_DAC_RID
Позволяет процессу задать или изменить свой ИД роли (RID)

Привилегии файловых систем:

В Trusted AIX существуют следующие привилегии файловых систем. Ниже приводится краткое описание каждой привилегии и ее применения. Некоторые привилегии образуют иерархию, в которой одна привилегия может предоставлять все права, связанные с другой привилегией.

При выяснении привилегий система сначала проверяет, есть ли у процесса наименьшая необходимая привилегия, а затем движется вверх по иерархии, проверяя наличие более широких привилегий. Например,

процесс с привилегией **PV_AU_** автоматически получает привилегии **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** и **PV_AU_WRITE**, а процесс с привилегией **PV_ROOT** - все перечисленные ниже привилегии, кроме привилегий **PV_SU_**.

PV_FS_
Эквивалентна объединению всех остальных привилегий **PV_FS_**

PV_FS_MKNOD
Позволяет процессу выполнить системный вызов **mknod** для создания файла любого типа.

PV_FS_MOUNT
Позволяет процессу смонтировать и размонтировать файловую систему

PV_FS_CHOWN
Позволяет процессу изменить принадлежность файла

PV_FS_QUOTA
Позволяет процессу управлять информацией о дисковых квотах

PV_FS_LINKDIR
Позволяет процессу создать жесткую ссылку на каталог

PV_FS_RESIZE
Позволяет процессу выполнять операции расширения и сокращения над файловой системой

PV_FS_CNTL
Позволяет процессу выполнять различные управляющие операции, кроме расширения и сокращения, над файловыми системами

PV_FS_CHROOT
Позволяет пользователю изменить свой корневой каталог

PV_FS_PDMODE
Позволяет процессу создать или задать разделенный каталог

Привилегии процессов:

В Trusted AIX существуют следующие привилегии процессов. Ниже приводится краткое описание каждой привилегии и ее применения. Некоторые привилегии образуют иерархию, в которой одна привилегия может предоставлять все права, связанные с другой привилегией.

При выяснении привилегий система сначала проверяет, есть ли у процесса наименьшая необходимая привилегия, а затем движется вверх по иерархии, проверяя наличие более широких привилегий. Например, процесс с привилегией **PV_AU_** автоматически получает привилегии **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** и **PV_AU_WRITE**, а процесс с привилегией **PV_ROOT** - все перечисленные ниже привилегии, кроме привилегий **PV_SU_**.

PV_PROC_
Эквивалентна объединению всех остальных привилегий **PV_PROC_**

PV_PROC_PRIO
Позволяет процессу или нити изменить приоритет, стратегию и другие параметры планирования

PV_PROC_CORE
Позволяет процессу создать дамп ядра

PV_PROC_RAC
Позволяет процессу создать больше процессов, чем разрешено одному пользователю

PV_PROC_RSET
Позволяет подключить набор ресурсов (**rset**) к процессу или нити

PV_PROC_ENV
Позволяет процессу задать пользовательскую информацию в пользовательской структуре

PV_PROC_CKPT
Позволяет процессу создать контрольную точку или перезапустить другой процесс

PV_PROC_CRED
Позволяет процессу задать атрибуты разрешений процесса

PV_PROC_SIG
Позволяет процессу отправить сигнал не связанному с ним процессу

PV_PROC_PRIV
Позволяет процессу изменить или просмотреть наборы привилегий, связанные с другим процессом

PV_PROC_TIMER
Позволяет процессу передать на выполнение и применить таймеры тонкой детализации

PV_PROC_RTCLK
Позволяет процессу получить доступ к часам CPU

PV_PROC_VARS
Позволяет процессу извлечь и обновить свои настраиваемые параметры

PV_PROC_PDMODE
Позволяет процессу изменить реальный режим разделенного каталога

Привилегии ядра:

В Trusted AIX существуют следующие привилегии ядра. Ниже приводится краткое описание каждой привилегии и ее применения. Некоторые привилегии образуют иерархию, в которой одна привилегия может предоставлять все права, связанные с другой привилегией.

При выяснении привилегий система сначала проверяет, есть ли у процесса наименьшая необходимая привилегия, а затем движется вверх по иерархии, проверяя наличие более широких привилегий. Например, процесс с привилегией **PV_AU_** автоматически получает привилегии **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** и **PV_AU_WRITE**, а процесс с привилегией **PV_ROOT** - все перечисленные ниже привилегии, кроме привилегий **PV_SU_**.

PV_KER_
Эквивалентна объединению всех остальных привилегий **PV_KER_**

PV_KER_ACCT
Позволяет процессу выполнять служебные операции, относящиеся к подсистеме учетных записей

PV_KER_DR
Позволяет процессу запускать операции динамической перенастройки

PV_KER_TIME
Позволяет процессу изменять часы и время системы

PV_KER_RAC
Позволяет процессу использовать большие (нефрагментируемые) страницы для сегментов общей памяти

PV_KER_WLM
Позволяет процессу инициализировать и изменить конфигурацию WLM

PV_KER_EWLM
Позволяет процессу инициализировать или запросить среду eWLM

PV_KER_VARS
Позволяет процессу изучить или задать настраиваемые параметры среды выполнения ядра

PV_KER_REBOOT
Позволяет процессу завершить работу системы

PV_KER_RAS
Позволяет процессу настроить или создать записи RAS, ведение протокола ошибок, трассировку и дамп

PV_KER_LVM
Позволяет процессу настроить подсистему LVM

PV_KER_NFS
Позволяет процессу настроить подсистему NFS

PV_KER_VMM
Позволяет процессу изменить параметры swap и другие настраиваемые параметры VMM в ядре

PV_KER_WPAR
Позволяет процессу настроить раздел рабочей нагрузки

PV_KER_CONF
Позволяет процессу выполнять различные операции по настройке системы

PV_KER_EXTCONF
Позволяет процессу выполнять различные задачи настройки в расширениях ядра

PV_KER_IPC
Позволяет процессу увеличить значение буфера очереди сообщений IPC и разрешить системные вызовы **shmget** с диапазонами подключения

PV_KER_IPC_R
Позволяет процессу прочесть очередь сообщений IPC, набор семафоров или сегмент общей памяти

PV_KER_IPC_W
Позволяет процессу записать очередь сообщений IPC, набор семафоров или сегмент общей памяти

PV_KER_IPC_O
Позволяет процессу прочесть и переопределить принадлежность DAC во всех объектах IPC

PV_KER_SECCONFIG
Позволяет процессу задать флаги защиты ядра

PV_KER_PATCH
Позволяет процессу вставлять расширения ядра

Привилегии меток:

В Trusted AIX существуют следующие привилегии меток. Ниже приводится краткое описание каждой привилегии и ее применения. Некоторые привилегии образуют иерархию, в которой одна привилегия может предоставлять все права, связанные с другой привилегией.

При выяснении привилегий система сначала проверяет, есть ли у процесса наименьшая необходимая привилегия, а затем движется вверх по иерархии, проверяя наличие более широких привилегий. Например, процесс с привилегией **PV_AU_** автоматически получает привилегии **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** и **PV_AU_WRITE**, а процесс с привилегией **PV_ROOT** - все перечисленные ниже привилегии, кроме привилегий **PV_SU_**.

PV_LAB_
Эквивалентна объединению всех остальных привилегий меток (**PV_LAB_***)

PV_LAB_CL
Позволяет процессу изменить SCL субъектов, с учетом допуска процесса

PV_LAB_CLTL
Позволяет процессу изменить TCL субъектов, с учетом допуска процесса

PV_LAB_LEF
Позволяет процессу считывать базу данных меток

PV_LAB_SLDG
Позволяет процессу понизить SL, с учетом допуска процесса

PV_LAB_SLDG_STR
Позволяет процессу понизить SL пакета, с учетом допуска процесса

PV_LAB_SL_FILE
Позволяет процессу изменить SL объектов, с учетом допуска процесса

PV_LAB_SL_PROC
Позволяет процессу изменить SL субъектов, с учетом допуска процесса

PV_LAB_SL_SELF
Позволяет процессу изменить свой собственный SL, с учетом допуска процесса

PV_LAB_SLUG
Позволяет процессу повысить SL, с учетом допуска процесса

PV_LAB_SLUG_STR
Позволяет процессу повысить SL пакета, с учетом допуска процесса

PV_LAB_TL
Позволяет процессу изменить TL субъектов и объектов

Привилегии MAC:

В Trusted AIX существуют следующие привилегии MAC. Ниже приводится краткое описание каждой привилегии и ее применения. Некоторые привилегии образуют иерархию, в которой одна привилегия может предоставлять все права, связанные с другой привилегией.

При выяснении привилегий система сначала проверяет, есть ли у процесса наименьшая необходимая привилегия, а затем движется вверх по иерархии, проверяя наличие более широких привилегий. Например, процесс с привилегией **PV_AU_** автоматически получает привилегии **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** и **PV_AU_WRITE**, а процесс с привилегией **PV_ROOT** - все перечисленные ниже привилегии, кроме привилегий **PV_SU_**.

PV_MAC_
Эквивалентна объединению всех остальных привилегий MAC (**PV_MAC_***)

PV_MAC_CL
Позволяет процессу обойти ограничения допуска секретности

PV_MAC_R_PROC
Позволяет процессу обойти ограничения на чтение MAC при получении информации о другом процессе, при условии что метка целевого процесса входит в допуск данного процесса

PV_MAC_W_PROC
Позволяет процессу обойти ограничения на запись MAC при отправке сигнала процессу, при условии что метка целевого процесса входит в допуск данного процесса

PV_MAC_R
Позволяет процессу обойти ограничения на чтение MAC

PV_MAC_R_CL
Позволяет процессу обойти ограничения на чтение MAC, при условии что метка объекта входит в допуск процесса

PV_MAC_R_STR
Позволяет процессу обойти ограничения на чтение MAC при чтении сообщения из STREAM, при условии что метка сообщения входит в допуск процесса

PV_MAC_W
Позволяет процессу обойти ограничения на запись MAC

PV_MAC_W_CL
Позволяет процессу обойти ограничения на запись MAC, при условии что метка объекта входит в допуск процесса

PV_MAC_W_DN
Позволяет процессу обойти ограничения на запись MAC, когда метка процесса поглощает метку объекта и метка объекта входит в допуск процесса

PV_MAC_W_UP
Позволяет процессу обойти ограничения на запись MAC, когда метка процесса поглощается меткой объекта и метка объекта входит в допуск процесса

PV_MAC_OVRRD
Позволяет обойти ограничения MAC для файлов, помеченных как свободные от MAC

Привилегии MIC:

В Trusted AIX существуют следующие привилегии MIC. Ниже приводится краткое описание каждой привилегии и ее применения. Некоторые привилегии образуют иерархию, в которой одна привилегия может предоставлять все права, связанные с другой привилегией.

При выяснении привилегий система сначала проверяет, есть ли у процесса наименьшая необходимая привилегия, а затем движется вверх по иерархии, проверяя наличие более широких привилегий. Например, процесс с привилегией **PV_AU_** автоматически получает привилегии **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** и **PV_AU_WRITE**, а процесс с привилегией **PV_ROOT** - все перечисленные ниже привилегии, кроме привилегий **PV_SU_**.

PV_MIC
Позволяет процессу обойти ограничения целостности

PV_MIC_CL
Позволяет процессу обойти ограничения допуска целостности

Сетевые привилегии:

В Trusted AIX существуют следующие сетевые привилегии. Ниже приводится краткое описание каждой привилегии и ее применения. Некоторые привилегии образуют иерархию, в которой одна привилегия может предоставлять все права, связанные с другой привилегией.

При выяснении привилегий система сначала проверяет, есть ли у процесса наименьшая необходимая привилегия, а затем движется вверх по иерархии, проверяя наличие более широких привилегий. Например, процесс с привилегией **PV_AU_** автоматически получает привилегии **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** и **PV_AU_WRITE**, а процесс с привилегией **PV_ROOT** - все перечисленные ниже привилегии, кроме привилегий **PV_SU_**.

PV_NET_
Эквивалентна объединению всех остальных сетевых привилегий (**PV_NET_***)

PV_NET_CNTL
Позволяет процессу изменять таблицы сети

PV_NET_PORT
Позволяет процессу подключиться к служебному порту

PV_NET_RAWSOCK
Позволяет процессу получить прямой доступ к сетевому слою

PV_NET_CONFIG
Позволяет процессу настроить параметры сети

Привилегии администратора:

В Trusted AIX существуют следующие привилегии администратора. Ниже приводится краткое описание каждой привилегии и ее применения. Некоторые привилегии образуют иерархию, в которой одна привилегия может предоставлять все права, связанные с другой привилегией.

При выяснении привилегий система сначала проверяет, есть ли у процесса наименьшая необходимая привилегия, а затем движется вверх по иерархии, проверяя наличие более широких привилегий. Например, процесс с привилегией **PV_AU_** автоматически получает привилегии **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** и **PV_AU_WRITE**, а процесс с привилегией **PV_ROOT** - все перечисленные ниже привилегии, кроме привилегий **PV_SU_**.

PV_SU_
Эквивалентна объединению всех остальных привилегий администратора (**PV_SU_***)

PV_SU_ROOT
Предоставляет процессу эквивалент всех привилегий, связанных со стандартным администратором

PV_SU_EMUL
Предоставляет процессу эквивалент всех привилегий, связанных со стандартным администратором, когда UID процесса равен 0

PV_SU_UID
Указывает, что системный вызов **getuid** должен возвратить 0

Различные привилегии:

В Trusted AIX существуют следующие различные привилегии. Ниже приводится краткое описание каждой привилегии и ее применения. Некоторые привилегии образуют иерархию, в которой одна привилегия может предоставлять все права, связанные с другой привилегией.

При выяснении привилегий система сначала проверяет, есть ли у процесса наименьшая необходимая привилегия, а затем движется вверх по иерархии, проверяя наличие более широких привилегий. Например, процесс с привилегией **PV_AU_** автоматически получает привилегии **PV_AU_ADMIN**, **PV_AU_ADD**, **PV_AU_PROC**, **PV_AU_READ** и **PV_AU_WRITE**, а процесс с привилегией **PV_ROOT** - все перечисленные ниже привилегии, кроме привилегий **PV_SU_**.

PV_ROOT
Предоставляет процессу эквивалент всех остальных привилегий, кроме **PV_SU_** (и привилегий, поглощаемых **PV_SU_**)

PV_TCB
Позволяет процессу изменять надежные пути к библиотеке ядра

PV_TP
Указывает, что процесс является процессом надежного пути, и позволяет выполнять действия, разрешенные только таким процессам

PV_TP_SET
Позволяет процессу задать или сбросить флаг надежного пути ядра

PV_WPAR_CKPT
Позволяет процессу проверить контрольную точку и перезапустить операции в разделах рабочей нагрузки

PV_DEV_CONFIG
Позволяет процессу настроить расширения ядра и устройства системы

PV_DEV_LOAD
Позволяет процессу загрузить или выгрузить расширения ядра и устройства системы

PV_DEV_QUERY
Позволяет процессу запросить модули ядра

Устранение неполадок Trusted AIX

В этом разделе описаны типичные способы устранения неполадок Trusted AIX.

Каким образом осуществляется вход в систему Trusted AIX?
В ходе установки Trusted AIX создаются три администратора с соответствующими ролями, описанными далее.

Пароли этих учетных записей указываются при первой загрузке системы после установки Trusted AIX. Если установка системы выполнялась неинтерактивно по сети, то пароли для этих учетный записей устанавливаются по умолчанию. Они перечислены ниже.

Пользователь	Пароль
isso	isso
sa	sa
so	so

Как выполнить команду su `root`?
Во время установки Trusted AIX атрибуту **su** для `root` присваивается значение **false**, чтобы доступ к этой учетной записи был закрыт для всех пользователей. Для разблокирования учетной записи root администраторы isso и sa должны задать этот атрибут равным **true** с помощью команды **chuser**.

Если включена возможность команды **su** root, а пароль для root не задан, то любой пользователь в системе будет иметь доступ к учетной записи root. Рекомендуется во избежание этого задать пароль для root перед изменением атрибута **su**

Следует ли создать собственных администраторов или достаточно администраторов по умолчанию?
Администраторы по умолчанию используются только для перенастройки системы. Настоятельно рекомендуется (хотя это необязательно) использовать эти учетные записи только для этих целей.

Создайте три администратора с соответствующими ролями isso, sa и so и удалите или выключите администраторов по умолчанию.

Почему невозможно войти в систему?
Вход в систему с учетной записью root (uid 0) или с любым uid меньше 128 невозможен. Эти учетные записи называются учетными записями системы. Для доступа к этим учетным записям войдите в систему как обычный пользователь и затем используйте команду **su** для смены учетной записи.

Какая ошибка связана с тем, что файл LabelEncodings показывается при входе в систему?
Если файл LabelEncodings поврежден, то необходимо войти в систему в однопользовательском режиме как пользователь root. Вход с учетной записью root возможен только в однопользовательском режиме.

Проверить правильность формата файла кодировки меток (`/etc/security/enc/LabelEncodings`) можно командой **labck**. Если файл поврежден, восстановите его, проверьте его еще раз командой **labck** и выйдите из однопользовательского режима.

Выполните команду **trustchk** в интерактивном режиме (**trustchk -t ALL**) для проверки состояния системы.

Почему не удается скомпилировать другую программу в Trusted AIX, которая использует библиотечный API Trusted AIX?
По умолчанию инструментарий для разработки не установлен. Необходимо установить набор файлов `bos.mls.adt` с установочного носителя.

Как исправить изменения, сделанные для прав доступа и приведшие к тому, что команды больше не работают?
Выполните команду **trustchk** в интерактивном режиме (**trustchk -t**) для исправления прав доступа.

Почему нет доступа к каталогу /etc/security/enc?

Обращение к каталогу /etc/security/enc требует прав доступа PV_LAB_LEF и PV_MAC_R. Предоставьте эти права доступа своей оболочке.

Как выключить команду trustchk во время загрузки?

Удалите или закомментируйте строку trustchk в сценарии /etc/rc.mls.

Как отменить запрос на идентификацию при каждой загрузке?

Вероятно, вы включили идентификацию при загрузке системы. Ее можно выключить в меню SMIT из меню Trusted Trusted AIX.

Почему не работают изменения SL для объекта файловой системы?

Возможны следующие причины:

Возвращает ли /usr/sbin/settxattr сообщение об ошибке?

Если да, то просмотрите их. Например:

Есть ли права доступа для выполнения /usr/sbin/settxattr?

Если нет, проверьте свои права доступа.

Была ли правильно вызвана команда?

Проверьте правильность формата вызове в справке по **settxattr**.

Существует ли запрошенная SL или ее сокращение?

Запрос "con a b" будет работать в системе с файлом LabelEncodings по умолчанию (/etc/security/enc/LabelEncodings), но запрос "conf a b" работать не будут, даже если оба эти сокращения выглядят похоже для выражения "confidential compartment A compartment B."

Указаны ли кавычки для метки с несколькими словами?

settxattr -f sl=con *<файл>* будет работать, settxattr -f -a sl="con a b" *<файл>* будет работать, но settxattr -a sl=con a b *<файл>* не будет работать.

Возвращает ли settxattr сообщение об ошибке?

Если нет сообщений об ошибке, то объект файловой системы может быть ссылкой. Если изменяемый объект - это ссылка, то сначала определите, что требуется изменить, SL самой ссылки или объект, на который она указывает. **settxattr** не следует по ссылкам, а изменяет метки самой ссылки.

Как установить сторонние приложения, чтобы они правильно работали в системе?

Если сторонние приложения работают неверно, возможно, что они обращаются к некоторым файлам или каталогам, к которым требуются дополнительные права доступа. Определив, к каким объектам приложению требуется доступ, затем определите, какие для этого нужны права доступа, как показано ниже.

- Присвойте своей оболочке права PV_ROOT
- Выполните команду tracepriv -f -e *<стороннее приложение>*

Эта команда покажет, какие права доступа необходимы приложению. Добавьте эти права доступа в базу данных привилегированных команд с помощью команды **setsecattr**.

Почему невозможно выполнение некоторых команд?

Большинство команд выполняется с соответствующими правами доступа, и поэтому для выполнения некоторых привилегированных команд необходимо, чтобы пользователь обладал такими правами. Можно проверить, предоставлены ли права, требуемые для выполнения команды, одной из ролей, активных для текущего сеанса.

Выполните команду rolelist -ae для показа своих активных прав доступа и lssecattr -c *<команда>* для показа необходимых прав доступа.

Почему некоторые команды неверно показывают метки?
Большинство команд выполняют преобразование меток из машинного вида в удобочитаемый и обратно на основе файла `/etc/security/enc/LabelEncodings`. Если этот файл поврежден или изменен, то работа команд может быть нарушена.

Флаги защиты файлов

Флаги защиты файлов влияют на способ доступа к файлам. Эти флаги хранятся как часть расширенных атрибутов (EA) самого файла. Флаги защиты файлов задаются в файле заголовка.

FSF_APPEND
Файлы можно только пополнять, но не изменять в рабочем режиме.

FSF_AUDIT
Файл помечен как часть подсистемы контроля. Для чтения этих файлов или записи в них у процесса должны быть привилегии **PV_AU_READ** или **PV_AU_WRITE** соответственно.

FSF_MAC_EXMPT
EPS с привилегией **PV_MAC_OVRRD** игнорирует ограничения MAC при попытке доступа к объекту.

FSF_PDIR
Каталог является разделенным каталогом.

FSF_PSDIR
Каталог является разделенным подкаталогом.

FSF_PSSDIR
Каталог является разделенным подподкаталогом.

FSF_TLIB
Объект помечен как часть Надежной библиотеки. Компьютер должен работать в режиме настройки или флаг защиты ядра **trustedlib_enabled** должен быть отключен.

FSF_TLIB_PROC
Процессы, помеченные как TLIB, могут связываться только с библиотеками ***.so**, у которых задан флаг **TLIB**. Система должна работать в режиме настройки или флаг защиты ядра **trustedlib_enabled** должен быть отключен.

Команды Trusted AIX

В этом разделе описаны команды, предназначенные для управления защитой Trusted AIX:

labck Проверяет файл LabelEncodings

getsecconf
Показывает флаги защиты ядра

setsecconf
Изменяет флаги защиты ядра Trusted AIX

getsyslab
Показывает максимальные и минимальные метки ядра

setsyslab
Задает максимальные и минимальные метки ядра

getrunmode
Показывает текущий рабочий режим системы

setrunmode
Переключает рабочий режим системы

pdlink Создает связи файлов в разделенных подкаталогах

pdmkdir
Создает разделенные каталоги и подкаталоги

pdmode
Возвращает текущий режим доступа к разделенным каталогам или выполняет команду с указанным режимом доступа к разделенному каталогу

pdrmdir
Удаляет разделенные каталоги и связанные подкаталоги

pdset Задает или отменяет разделенные (под)каталоги

bootauth
Проверяет, загружает ли систему пользователь с соответствующими правами доступа

chuser Изменяет атрибуты допуска пользователя

lsuser Показывает атрибуты допуска пользователя

chsec Изменяет атрибуты допуска пользователя и метки портов

lssec Показывает атрибуты допуска пользователя и метки портов

trustchk
Проверяет атрибуты файлов

lstxattr
Показывает метки и атрибуты флагов защиты файлов, процессов и объектов IPC

settxattr
Изменяет метки и атрибуты флагов защиты файлов, процессов и объектов IPC

Примечания

Данная информация была разработана для продуктов и услуг, предлагаемых на территории США.

Компания IBM может не предоставлять в других странах продукты и услуги, обсуждаемые в данном документе. Информацию о продуктах и услугах, распространяемых в вашей стране, вы можете получить в местном представительстве IBM. Ссылки на продукты, программы или услуги IBM не означают, что можно использовать только указанные продукты, программы или услуги IBM. Вместо них можно использовать любые другие функционально эквивалентные продукты, программы или услуги, не нарушающие прав IBM на интеллектуальную собственность. Однако ответственность за проверку действия любых продуктов, программ и услуг других компаний лежит на пользователе.

Компания IBM может обладать заявками на патенты или патентами на предметы обсуждения в данном документе. Обладание данным документом не предоставляет лицензии на эти патенты. Запросы на получение лицензии можно отправлять в письменном виде по адресу:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

За получением лицензий, имеющих отношение к двухбайтовому набору символов (DBCS), обращайтесь в местное отделение компании IBM по интеллектуальной собственности или направьте запрос в письменной форме по следующему адресу:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

КОМПАНИЯ IBM ПРЕДОСТАВЛЯЕТ НАСТОЯЩУЮ ПУБЛИКАЦИЮ НА УСЛОВИЯХ "КАК ЕСТЬ", БЕЗ КАКИХ-ЛИБО ЯВНЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ ГАРАНТИЙ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ЭТИМ, НЕЯВНЫЕ ГАРАНТИИ СОБЛЮДЕНИЯ ПРАВ, КОММЕРЧЕСКОЙ ЦЕННОСТИ И ПРИГОДНОСТИ ДЛЯ КАКОЙ-ЛИБО ЦЕЛИ. В некоторых юрисдикциях освобождение от явных и подразумеваемых гарантий запрещено в некоторых сделках, поэтому это заявление может к вам не относиться.

Эта информация может содержать технические неточности или типографические ошибки. В информацию периодически вносятся изменения, которые будут учтены во всех последующих изданиях этой книги. IBM может вносить обновления или изменения в этот документ без предварительного уведомления.

Любые ссылки на веб-сайты других компаний приведены в данной публикации исключительно для удобства пользователей и не должны рассматриваться как рекомендация этих веб-сайтов. Материалы, размещенные на этих веб-сайтах, не являются частью информации по данному продукту IBM, и ответственность за применение этих материалов лежит на пользователе.

IBM может использовать и распространять предоставленную вами информацию любым способом без каких-либо обязательств перед вами.

Лицам, обладающим лицензией на данную программу и желающим получить информацию о ней с целью: (i) настройки обмена данными между независимо разработанными программами и другими программами (включая данную) и (ii) использования информации, полученной в результате обмена, этими программами, следует обращаться по адресу:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Такая информация может быть предоставлена на определенных условиях, а в некоторых случаях - и за дополнительную плату.

Описанная в этом документе лицензионная программа и все связанные с ней лицензионные материалы предоставляются IBM в соответствии с условиями Соглашения с заказчиком IBM, Международного соглашения о лицензии на программу IBM или любого другого эквивалентного соглашения.

Данные о производительности и примеры клиентов приведены исключительно в иллюстративных целях. Фактические результаты производительности зависят от конкретных конфигураций и рабочих сред.

Информация о продуктах других компаний была получена от поставщиков этих продуктов, их опубликованных материалов или других общедоступных источников. Компания IBM не проверяла эти продукты и не может подтвердить правильность их работы, совместимость или другие заявленные характеристики продуктов других компаний. По вопросам о возможностях продуктов других компаний следует обращаться к поставщикам этих продуктов.

Заявления относительно будущих намерений IBM могут быть изменены или отозваны без дополнительного уведомления и отражают только текущие цели и задачи.

Все указанные цены IBM являются рекомендуемыми розничными ценами IBM на данный момент и могут быть изменены без предварительного уведомления. Цены дилеров могут быть другими.

Данная информация предназначена только для планирования. Она может быть изменена до выпуска описанных в данном документе продуктов.

Настоящая документация содержит примеры данных и отчетов, применяемых в повседневной деятельности компаний. Для большего сходства с реальностью примеры содержат имена людей, названия компаний, товарных знаков и продуктов. Все эти имена и названия вымышленные. Любые совпадения с реально существующими физическими или юридическими лицами совершенно случайны.

Лицензия на авторские права:

Настоящая документация содержит примеры исходного кода программ, иллюстрирующие приемы программирования в различных операционных системах. Вы имеете право копировать, изменять и распространять эти примеры программ в любой форме без уплаты вознаграждения фирме IBM в целях разработки, применения, сбыта или распространения прикладных программ, соответствующих интерфейсу прикладных программ операционной системы, для которой предназначены эти примеры. Эти примеры не были тщательно и всесторонне протестированы. В связи с этим IBM не может гарантировать их надежность, удобство обслуживания и отсутствие ошибок. Примеры программ предоставляются "КАК ЕСТЬ", без каких-либо гарантий. IBM не несет ответственности за ущерб, который может возникнуть в результате использования эти образцов программ.

Во все копии или фрагменты этих примеров программ, а также программы созданные на их основе, следует добавлять следующее замечание об авторских правах:

© (название вашей компании) (год).

Некоторые фрагменты исходного кода получены из примеров программ фирмы IBM Corp.

© Copyright IBM Corp. _год или годы_.

Замечания о правилах работы с личными данными

Продукты IBM Software, включая решения программного обеспечения как услуг, (“Предложения программного обеспечения”) могут использовать cookie или другие технологии для сбора информации об использовании продукта в целях усовершенствования пользовательского интерфейса, для приспособления взаимодействий к конечному пользователю или для других целей. Во многих случаях Предложениями программного обеспечения собирается информация, в которой невозможно опознать персональные данные. Некоторые из наших Предложений программного обеспечения могут позволить вам собирать опознаваемую персональную информацию. Если это Предложение программного обеспечения использует cookie для сбора опознаваемой персональной информации, то специфическая информация об этом использовании cookie в предложении приведена далее.

Это Предложение программного обеспечения не использует cookie или другие технологии для сбора опознаваемой персональной информации.

Если конфигурации, развернутые для этого Предложения программного обеспечения предоставляют вам как клиенту возможность собирать опознаваемую персональную информацию о конечных пользователях посредством cookie и других технологий, вы должны самостоятельно проконсультироваться с юристом о всех законах, применимых к такому сбору данных, включая требования к уведомлению и согласию.

Более подробная информация об использовании различных технологий, включая cookie, для этих целей, приведена в Политике конфиденциальности IBM (http://www.ibm.com/privacy) и Заявлении IBM о конфиденциальности в Интернет (http://www.ibm.com/privacy/details), а также в разделах “Cookies, Web Beacons and Other Technologies” и “IBM Software Products and Software-as-a-Service Privacy Statement” на странице http://www.ibm.com/software/info/product-privacy.

Товарные знаки

IBM, эмблема IBM и ibm.com являются товарными знаками или зарегистрированными товарными знаками International Business Machines Corp. во всем мире. Названия других продуктов и услуг могут быть товарными знаками IBM и других компаний. Текущий список товарных знаков IBM опубликован на веб-странице Copyright and trademark information по следующему адресу: www.ibm.com/legal/copytrade.shtml.

Linux является зарегистрированным товарным знаком Линуса Торвальдса в США и других странах.

Microsoft и Windows NT являются товарными знаками фирмы Microsoft Corporation в США и других странах.

Java и все основанные на Java названия и эмблемы являются товарными знаками или зарегистрированными товарными знаками Oracle и/или дочерних компаний.

UNIX - зарегистрированный товарный знак The Open Group в США и других странах.

Индекс

Спец. символы

A

B

D

E

F

I

K

L

M

N

O

P

R

S

T

U

V

X

M

H

O

П

P

С

Т

У

Ф

Ц

Ш

Э

Напечатано в Дании