Глава 4

Уровень RHCSA Параметры безопасности

ЦЕЛИ СЕРТИФИКАЦИИ

4.01 Основные права доступа к файлам
4.02. Списки контроля доступа и многое другое
4.03 Базовый контроль брандмауэра
4.04 Защита SSH с помощью ключей
Аутентификация
4.05 Учебник по Linux с улучшенной безопасностью
✓ Двухминутный просмотр темы
Q&A Самостоятельный тест

Безопасность **Linux** начинается с концепции, известной как дискреционное управление доступом (**DAC**). Это включает в себя разрешения и права собственности, связанные с файлами и каталогами. С помощью специализированных битов, включая списки контроля доступа (**ACL**), разрешения могут быть более детализированными, чем простые категории **user/group/other**. **ACL-списки** поддерживают разрешения, предоставляемые определенным пользователям или группам, переопределяют стандартные разрешения и предоставляют более детальные правила доступа для данного файла.

Также в области безопасности находится межсетевой экран. В этой главе вы изучите как службу **iptables** (которая была брандмауэром по умолчанию в RHEL 6), так и новый демон **firewalld**, который обеспечивает поддержку **различных зон доверия**. Вы узнаете, как разрешать или блокировать службы через **firewalld**, используя графическую утилиту **firewall-config** и командный инструмент **firewall-cmd**.

Служба, которая установлена в большинстве систем **Linux - это SSH**. Поскольку это очень распространенный сервис для входа в систему, хакеры «черной шляпы» везде хотят **найти уязвимость в SSH**. Так что в этой главе описывается, как вы можете улучшить безопасность, используя аутентификацию **на основе ключей для SSH**.

Дополнительную защиту можно обеспечить с помощью другого типа защиты, известного как обязательное управление доступом (**MAC (mandatory access control)**). Реализация MAC RHEL 7 известна как Linux с улучшенной безопасностью (**SELinux (Enhanced Linux)**). Red Hat ожидает, что вы будете работать с включенным (**enforcing**) **SELinux** во время экзаменов. В этой главе вы узнаете, как устанавливать режимы принудительного применения, изменять контексты файлов, использовать логические параметры и диагностировать нарушения политики SELinux.

Если вы начинаете с установки по умолчанию, созданной в процессе установки, вам может потребоваться установить дополнительные пакеты в этой главе. Если удаленный репозиторий доступен, возьмите имя пакета и примените к нему команду **yum install**. Например, чтобы просмотреть инструмент настройки брандмауэра на основе графического интерфейса, вам необходимо установить его с помощью следующей команды:

 # yum install firewall-config

Для получения дополнительной информации о процессе установки пакета см. **Главу 7**.

ВНУТРИ ЭКЗАМЕНА

Основные права доступа к файлам

Безопасность в Linux начинается с прав доступа к файлам. Поскольку все в Linux можно определить как файл, это отличное начало. В любом случае, связанные с этим цели, будучи понятыми, довольно просты:

- Перечислите, установите и измените стандартные разрешения **ugo/rwx(users,groups,others/raid,write,execute)**
- Диагностика и исправление проблем с правами доступа к файлам

Стандартные разрешения для файлов Linux определены **для пользователей, групп и других**, сокращённо **ugo**. Разрешения - **чтение, запись и выполнение**, сокращённо определенны, как **rwx**.

Такие разрешения как дискреционное управление доступом, в отличие от системы обязательного контроля доступа, известной, как **SELinux**, также обсуждаемой в этой главе.

Списки контроля доступа

ACL могут быть настроены для переопределения и расширения основных прав доступа к файлам. Например, **с помощью ACL** вы можете настроить файл в своем домашнем каталоге, который будет доступен для чтения ограниченному числу других пользователей и групп. Соответствующая задача RHCSA

- Создание и управление списками контроля доступа (ACL)

Управление брандмауэром

Как настроено в Linux, брандмауэр может блокировать трафик на всех, кроме нескольких сетевых портов. Он также может быть использован для регулирования дорожного движения рядом других способов, но это является областью проведения экзамена RHCE. Соответствующая цель экзамена RHCSA

- Настройте параметры брандмауэра с помощью firewall-config, firewall-cmd или iptables

Secure Shell Server

Как указано во введении, особое внимание уделяется **услуге SSH**. Соответствующая цель RHCSA

- Настройте аутентификацию на основе ключей для SSH

Благодаря аутентификации на основе ключей вы сможете входить в удаленные системы, **используя пары секретных/открытых ключей**. Передача пароля по сети больше не требуется. 1024 или более битов, связанных с такой аутентификацией, взломать намного сложнее, чем пароль, передаваемый по сети.

Linux с улучшенной безопасностью

Нет никакого способа обойти **SELinux**. На экзаменах **Red Hat** вы должны работать с **SELinux**. Неясно, сможете ли вы даже сдать экзамены **Red Hat**, если по крайней мере некоторые службы не настроены на **SELinux**. Чтобы помочь кандидатам на экзаменах понять, что необходимо, Red Hat разбила задачи, связанные с SELinux. Первая цель

является фундаментальной для **SELinux**, так как она относится к трем режимам, доступным для **SELinux** в системе (обеспечение соблюдения (**enforcing /permissive/disabled**):

- Установить принудительный (**enforcing**) и разрешающий (**permissive**) режимы для **SELinux**

Следующая цель требует, чтобы вы понимали **контексты SELinux**, определенные для разных файлов и процессов. Хотя связанные команды просты, доступные контексты столь же широки, как и количество сервисов, доступных в Linux:

- Перечисление и описания файлов **SELinux** и контексты процесса

Когда вы экспериментируете с разными **контекстами SELinux**, случаются ошибки. Вы можете не помнить контексты по умолчанию, связанные с важными каталогами. Но с правильными командами вам не нужно помнить все; Как следует из следующей задачи, восстановить значение по умолчанию относительно легко:

- Восстановить контексты файлов по умолчанию

Следующая цель может показаться сложной. Но **логические (boolean) настройки**, связанные с **SELinux**, имеют описательные имена. Доступны отличные инструменты для дальнейшего уточнения доступных логических контекстов. По сути, это означает, что для запуска определенной службы под **SELinux** все, что вам нужно сделать, это включить один или несколько логических параметров (вместо того, чтобы изменять правила политики **SELinux** напрямую):

- Используйте **логические (boolean)** настройки для изменения системных настроек **SELinux**

После того, как **SELinux** заработал, вы должны следить за нарушениями политики системы. Нарушение может быть результатом неправильной конфигурации или несанкционированной попытки вторжения. Следовательно, чтобы получить максимальную отдачу от **SELinux**, вы должны знать, как проводить аудит на предмет нарушений правил, и уметь решать распространенные проблемы. Соответствующая цель RHCSA

- Диагностика и устранение регулярных, обычных нарушений **политики SELinux**.

ЦЕЛЬ СЕРТИФИКАЦИИ 4.01

Основные права доступа к файлам

Базовая безопасность компьютера Linux основана на правах доступа к файлам. Права доступа к файлу по умолчанию устанавливаются с помощью команды **umask**. Специальные разрешения могут быть настроены для предоставления всем пользователям и/или группам дополнительных прав. Они известны как идентификатор суперпользователя (**SUID**), идентификатор супергруппы (**SGID**) и фиксированные биты разрешения. Принадлежность, основано на идентификаторах пользователя и группы по умолчанию для человека, который создал файл. Управление разрешениями и владением включает такие команды, как **chmod**, **chown** и **chgrp**. Прежде чем исследовать эти команды, важно понять права доступа и владельца, связанные с файлом.

Права доступа к файлам и право собственности

Права доступа к файлам **Linux** и их владение просты. Как следует из соответствующей цели RHCSA, они читают, записывают и выполняют, классифицированные пользователем, группой и всеми другими пользователями. Однако влияние разрешений на каталоги более тонкое. **Таблица 4-1** показывает точное значение каждого бита разрешения.

ТАБЛИЦА 4-1 Разрешения на файлы и Директории

Разрешение	На файл	На каталог
Читать ride (r)	Разрешение на чтение файла	Разрешение на перечисление содержимого каталога
Записывать write (w)	Разрешение на запись (изменение) файла	Разрешение на создание и удаление файлов в каталоге
Выполнить execute (x)	Разрешение на запуск файла как программы	Разрешение на доступ к файлам в каталоге

ТАБЛИЦА 4-2 Описание прав доступ файла

Позиция	Описание
1	Тип файла; - = обычный файл, ***d*** = каталог, ***b*** = устройство, ***l*** = символическая ссылка
234	Права, предоставленные владельцу файла
567	Права, предоставленные владельцу группы для файла
890	Разрешения, предоставленные всем другим пользователям в системе Linux

Рассмотрим следующий вывод из **ls -l /sbin/fdisk**:

-rwxr-xr-x. 1 root root 182424 Mar 28 2014 /sbin/fdisk

Права доступа указаны в левой части списка. Десять символов показаны. Первый символ определяет, является ли это обычный или специальный файл. Оставшиеся девять символов сгруппированы по три, в зависимости от владельца файла (пользователя), владельца группы и всех остальных в этой системе Linux. Буквы просты: **r = чтение, w = запись, x = выполнить**. Эти разрешения описаны в **таблице 4-2**.

Обычно владельцы файлов и пользователи групп имеют одинаковые имена. В этом случае пользователь **root** является членом группы **root**. Но им не обязательно иметь одно и то же имя. Например, каталоги, предназначенные для совместной работы пользователей, могут принадлежать специальной группе. Как, будет обсуждаться в **Главе 8**, это включает группы с несколькими обычными пользователями в качестве участников.

Помните, что разрешения, предоставленные группе, имеют приоритет над разрешениями, предоставленными всем другим пользователям. Точно так же разрешения, предоставленные владельцу, имеют приоритет над всеми другими категориями разрешений. Таким образом, в следующем примере, хотя у всех остальных есть полные права доступа к файлу, членам группы «**mike**» не были предоставлены какие-либо разрешения, и поэтому они не смогут читать, изменять или выполнять файл:

$ ls -l setup.sh
-rwx---rwx. 1 root mike 127 Dec 13 07:21 setup.sh

Есть относительно новый элемент с разрешениями - и он тонкий. Обратите внимание на точку после последнего **x** в выводе команды **ls -l setup.sh**? Он указывает, что файл **имеет контекст безопасности SELinux**. Если вы настроили разрешения **ACL** для

файла, эта **точка заменяется знаком плюс (+)**. Но этот символ не отменяет управление SELinux.

Вы должны рассмотреть другой тип разрешения: биты специального разрешения. Это не только биты **SUID** и **SGID**, но и другое специальное разрешение, известный, как **липкий** бит **sticky**. Влияние битов специального разрешения на файлы и каталоги показано в **таблице 4-3**.

Пример бита **SUID** связан с командой **passwd** в каталоге **/usr/bin**. Команда **ls -l** для этого файла приводит к следующему выводу:

-rwsr-xr-x. 1 root root 27832 Jan 30 2014 /usr/bin/passwd

ТАБЛИЦА 4-3 Биты специального разрешения

Специальное разрешение	В исполняемом файле	В каталоге
SUID	Когда файл выполняется, эффективным идентификатором пользователя процесса является идентификатор файла.	Нет эффекта.
SGID	Когда файл выполняется, эффективный идентификатор группы процесса равен идентификатору файла.	Предоставьте файлам, созданным в каталоге, ту же группу, что и для каталога.
Sticky bit	Нет эффекта.	Файлы в каталоге могут быть переименованы или удалены только их владельцами

s в бите выполнения для владельца файла является битом **SUID**. Это означает, что файл может быть выполнен другими пользователями с полномочиями владельца файла, пользователя с правами администратора. Но это не значит, что любой пользователь может изменить пароли другого пользователя. Доступ к команде **passwd** дополнительно регулируется подключаемыми модулями аутентификации (**PAM**), как **описано в главе 10**. Это навык RHCE. Пример бита **SGID** можно найти с помощью команды **ssh-agent**, также в каталоге **/usr/bin**. Он имеет бит **SGID** для правильного хранения парольных фраз. Команда **ls -l** для этого файла отображает следующий вывод:

---x--s--x. 1 root nobody 145312 Mar 19 2014 /usr/bin/ssh-agent

s в бите выполнения для владельца группы файла (**group nobody**) - это бит **SGID**. Наконец, пример залипающего бита можно найти в разрешениях каталога **/tmp**. Это означает, что пользователи могут копировать свои файлы в этот каталог, но никто другой не может удалить эти файлы, кроме их соответствующих владельцев (которые являются «**липкими sticki**»). Команда **ls -ld** в этом каталоге показывает следующий вывод:

drwxrwxrwt. 22 root root 4096 Dec 15 17:15 /tmp

t в бите выполнения для других пользователей является **sticky** битом. Обратите внимание, что без **sticky bit** каждый сможет удалить файлы всех остальных в **/tmp**, потому что разрешения на запись были предоставлены всем пользователям в этом каталоге.

Лазейка в разрешениях на запись

Легко удалить разрешения на запись из файла. Например, если вы хотите сделать файл **license.txt** доступным только для чтения, следующая команда удаляет разрешения на запись из этого файла:

$ chmod a-w license.txt

Однако пользователь, которому принадлежит файл, все еще может вносить изменения. Он не будет работать в текстовых редакторах с графическим интерфейсом, таких? как **gedit**. Он даже не будет работать в текстовом редакторе **Nano**. Но если внести изменения в текстовом редакторе **vi**, пользователь, которому принадлежит этот файл, может переопределить отсутствие прав записи **с помощью символа восклицательного знака**, **(!).** Другими словами, находясь в редакторе **vi**, пользователь, которому принадлежит файл, может выполнить следующую команду, чтобы преодолеть отсутствие разрешений на запись:

w!

Хотя это может показаться удивительным, на практике **w!** Команда **редактора vi** не обходит систему доступа к файлам Linux. **w!** Команда перезаписывает файл, то есть удаляет существующий файл и создает новый с тем же именем. Как видно из **таблицы 4-1**, бит разрешения, который дает право создавать и удалять файлы, является разрешением на запись в родительский каталог, а не разрешением на запись в сам файл. Следовательно, если пользователь имеет разрешение на запись в каталог, он может перезаписывать файлы в нем, независимо от битов разрешения записи, установленных для файлов.

Команды для изменения прав доступа и владельца

Ключевые команды, которые могут помочь вам управлять разрешениями и владельцем файла: **chmod, chown** и **chgrp**. В следующих подразделах вы узнаете, как использовать эти команды для изменения разрешений вместе с пользователем и группой, которой принадлежит определенный файл или даже ряд файлов.

Один из советов, который может помочь вам изменить разрешения для ряда файлов, - **использовать ключ -R**. Это рекурсивный переключатель для всех трех команд. Другими словами, если вы укажете **ключ -R** с какой-либо из отмеченных команд в каталоге, он применяет изменения рекурсивно. Изменения применяются ко всем файлам в этом каталоге, включая все подкаталоги. Рекурсия означает, что изменения также применяются к файлам в каждом подкаталоге и т. д.

Команда chmod

Команда **chmod** использует числовое значение разрешений, связанных с владельцем, группой и другими. В Linux разрешениям присваиваются следующие числовые значения: ***r*** = 4, ***w*** = 2 и ***x*** = 1. В числовом формате разрешения представлены восьмеричным числом, где каждая цифра связана с другой группой разрешений. Например, **номер разрешения 640** означает, что владельцу назначено **разрешение 6 (чтение и запись)**, тогда как у группы **есть разрешение 4 (чтение)**, а у всех остальных нет разрешений. Команды **chown** и **chgrp** настраивают владельцев пользователей и групп, связанных с указаным файлом.

Команда **chmod** является гибкой. Вам не всегда нужно использовать цифры. Например, следующие команды устанавливают разрешения на выполнение для владельца файла **Ch3Lab1**:

 # chmod u + x Ch3Lab1

Обратите внимание, как **u** и **x** следуют **формату ugo/rwx**, указанному в соответствующей цели RHCSA. Для объяснения, эта команда добавляет (со знаком плюс) для пользователя-владельца файла (**с помощью u**) разрешения на выполнение (**с помощью x**).

Эти символы могут быть объединены. Например, следующая команда отключает (со знаком минус) права записи (с **помощью w**) для владельца группы (с **помощью g**) и всех других пользователей (с **помощью o**) в локальном файле с именем special:

 # chmod go-w special

Вместо добавления или удаления разрешений с помощью **операторов + и -** вы можете установить точный режим группы разрешений с помощью **оператора равенства (=).** Например, следующая команда изменяет групповые разрешения файла с именем **special** на чтение и запись и удаляет разрешение на выполнение, если оно было установлено:

 # chmod g=rw special

Хотя вы можете использовать все три типа групповых разрешений в команде **chmod**, в этом нет необходимости. Как описано в лабораторных работах в главе 3, следующая команда делает указанный файл исполняемым для всех пользователей:

 # chmod +x Ch3Lab2

Для SUID, SGID и липких битов доступны некоторые специальные опции. Если вы решите использовать числовые биты, этим специальным битам также будут назначены числовые значения, где **SUID = 4, SGID = 2 и sticky bit = 1**. Например, следующая команда конфигурирует бит SUID (с первым «4» цифра в режиме разрешения). Он включает разрешения rwx для владельца пользователя (с «7»), разрешения rw для владельца группы (с «6») и разрешения r для других пользователей (с последними «4») в файле с именем **testfile**:

 # chmod 4764 testfile

Если вы предпочитаете использовать **формат ugo/rwx**, следующая команда активирует бит **SGID** для локального файла тестового сценария:

 # chmod g+s testcript

И следующая команда включает **sticky бит** для каталога **/test**:

 # chmod o+t /test

Для команды **chmod** изменения не должны вноситься пользователем с правами администратора. Пользователь-владелец файла может изменять разрешения, связанные с его файлами.

Команда chown

Команда **chown** может использоваться для изменения пользователя, которому принадлежит файл. Например, взглянем на право собственности на первую фигуру, которую мы создали для этой главы, основываясь на команде **ls -l**:

-rw-r--r--. 1 michael examprep 855502 Oct 25 14:07 F04-01.tif

Владельцем этого файла является **michael**; владельцем группы этого файла является **examprep**. Следующая команда **chown** меняет владельца пользователя на пользователя **elizabeth**:

 # chown elizabeth F04-01.tif

Вы можете сделать с командой **chown** гораздо больше; например, следующая команда меняет как пользователя, так и владельца группы указанного файла на пользователя **donna** и группы **supervisors**, предполагая, что пользователь и группа уже существуют:

 # chown donna.supervisors F04-01.tif

Только пользователь с правами администратора может изменить владельца файла, в то время как владение группой может быть изменено, как пользователем root, так и пользователем, которому принадлежит файл.

Команда chgrp

Вы можете изменить владельца группы файла с помощью команды **chgrp**. Например, следующая команда меняет владельца группы указанного файла **F04-01.tif** на группу с именем **project** (при условии, что она существует):

 # chgrp project F04-01.tif

Специальные атрибуты файла

За пределами обычных прав доступа **rwx/ugo** находятся атрибуты файла. Такие атрибуты могут помочь вам контролировать то, что каждый может делать с разными файлами. В то время как команда **lsattr** выводит список текущих атрибутов файла, команда **chattr** может помочь вам изменить эти атрибуты. Например, следующая команда защищает **/etc/fstab** от случайного удаления даже пользователем с правами администратора **root**:

 # chattr + i /etc/fstab

С этим атрибутом, если вы попытаетесь удалить файл от имени пользователя root, вы получите следующий ответ:

 # rm /etc/fstab
rm: remove regular file '/etc/fstab'? y
rm: cannot remove '/etc/fstab': Operation not permitted

Команда **lsattr** показывает активный неизменяемый атрибут в **/etc/fstab**:

 # lsattr /etc/fstab
----i----------- /etc/fstab

Конечно, пользователь с правами администратора может сбросить этот атрибут с помощью следующей команды. Тем не менее, первоначальный отказ от удаления файла должен по крайней мере дать паузу этому администратору перед внесением изменений:

 # chattr -i /etc/fstab

Несколько ключевых атрибутов описаны в **таблице 4-4**. Другие атрибуты, такие как **c (compressed), s (secure deletion)** и **u (undeletable),** не работают с файлами, хранящимися в файловых системах **ext4** и **XFS**. Атрибут формата **extent** связан с системами **ext4**.

ТАБЛИЦА 4-4 атрибутов файла

Атрибут	Описание
append only (a)	Предотвращает удаление, но разрешает добавление в файл - например, если вы запустили **chattr +a tester, cat /etc/fstab >> tester** добавит содержимое **/etc/fstab** в конец файла **tester**. Однако команда **cat /etc/fstab > tester** не будет выполнена.
no dump (d)	Запрещает резервное копирование настроенного файла с помощью команды **dump**.
extent format (e)	Установить с файловой системой **ext4**; атрибут, который нельзя удалить.
immutable (i)	Предотвращает удаление или любые другие изменения файла.

Основные понятия пользователя и группы

Linux, как и Unix, настроен на пользователей и группы. Каждый, кто использует Linux, настроен с именем пользователя, даже если это просто «**guest**». Есть даже стандартный пользователь с именем «**nobody**». Посмотрите на **/etc/passwd**. Одна версия этого файла показана на **рисунке 4-1**.

Как показано, все виды имен пользователей перечислены в файле **/etc/passwd.** Даже некоторые сервисы Linux, такие как **mail, news, ftp** и **apache**, имеют свои собственные имена пользователей. В любом случае, файл **/etc/passwd** имеет определенный формат, более подробно описанный в **главе 8**. На данный момент обратите внимание, что единственными постоянными пользователями, показанными в этом файле, являются **alex** и **michael**; их идентификаторы пользователей (**UID**) и идентификаторы групп (**GID**) составляют соответственно **1000** и **1001**; и их домашние каталоги соответствуют их именам пользователей. Следующий пользователь получает **UID** и **GID 1002** и так далее.

Это сопоставление **UID** и **GID** основано на схеме частной группы пользователей Red Hat. Теперь запустите команду **ls -l /home**. Вывод должен быть похож на следующее:

drwx------. 4 alex alex 4096 Dec 15 16:12 alex
drwx------. 4 michael michael 4096 Dec 16 14:00 michael

Обратите внимание на разрешения. Исходя из концепций **rwx/ugo**, описанных ранее в этой главе, только владелец имени пользователя имеет доступ к файлам в своем домашнем каталоге.

umask

Способ работы umask в Red Hat Enterprise Linux может быть удивительным, особенно если вы работаете в другой среде в стиле Unix. Вы не можете настроить umask, чтобы разрешить автоматическое создание новых файлов с исполняемыми разрешениями. Это повышает безопасность: если меньше файлов имеют разрешения на выполнение, меньше файлов доступно для хакера «черной шляпы», который можно использовать для запуска программ, чтобы пробиться через вашу систему.

РИСУНОК 4-1 файл /etc/passwd

```
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
named:x:25:25:Named:/var/named:/sbin/nologin
oprofile:x:16:16:Special user account to be used by OProfile:/var/lib/oprofile:/
sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
colord:x:998:996:User for colord:/var/lib/colord:/sbin/nologin
abrt:x:173:173::/etc/abrt:/sbin/nologin
chrony:x:997:995::/var/lib/chrony:/sbin/nologin
libstoragemgmt:x:996:994:daemon account for libstoragemgmt:/var/run/lsm:/sbin/no
login
qemu:x:107:107:qemu user:/:/sbin/nologin
radvd:x:75:75:radvd user:/:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
saslauth:x:995:76:"Saslauthd user":/run/saslauthd:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
unbound:x:994:993:Unbound DNS resolver:/etc/unbound:/sbin/nologin
pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
gnome-initial-setup:x:993:991::/run/gnome-initial-setup/:/sbin/nologin
alex:x:1000:1000:Alessandro Orsaria:/home/alex:/bin/bash
michael:x:1001:1001:Michael Jang:/home/michael:/bin/bash
```

Каждый раз, когда вы создаете новый файл, разрешения по умолчанию основаны на значении **umask**. Когда вы вводите команду **umask**, команда возвращает восьмеричное число, такое, как 0002. Если бит **umask** установлен, то соответствующее разрешение отключается во вновь создаваемых файлах и каталогах. Например, значение **umask 0245** приведет к тому, что вновь созданные каталоги будут иметь восьмеричные разрешения **0532**, что эквивалентно следующей строке разрешений

r-x-wx-w-.

В прошлом значение **umask** влияло на значение всех прав доступа к файлу. Например, если значение **umask было 000**, разрешения по умолчанию для любого файла, созданного этим пользователем, когда-то были **777 - 000 = 777**, что соответствует разрешениям на чтение, запись и выполнение для всех пользователей. Сейчас их **666**, так как обычные новые файлы больше не могут получать разрешения на выполнение. Каталоги, с другой стороны, требуют исполняемых разрешений, чтобы к любому доступному файлу можно было получить доступ.

umask по умолчанию

Имея это в виду, **umask** по умолчанию управляется файлами **/etc/profile** и **/etc/bashrc**, в частности, следующим разделом, который задает значение для **umask** в зависимости от значения **UID**:

```
if [ $UID -gt 199 ] && [ "`id -gn`" = "`id -un`" ]; then
        umask 002
else
        umask 022
fi
```

Другими словами, значение **umask** для учетных записей пользователей с **UID** от **200** и выше равно **002**. Напротив, значение **umask** для **UID** ниже **200** составляет **022**. В RHEL 7 такие пользователи службы, как **adm**, **postfix** и **apache**, имеют более **низкие UID**; это влияет в первую очередь на разрешения файлов журнала, созданных для таких служб. Конечно, пользователь **root** с самым низким **UID равен 0**. По умолчанию файлы, созданные для таких пользователей, имеют **644** разрешения; Каталоги, созданные для таких пользователей, имеют **755** разрешений.

В отличие от обычных пользователей, **UID 1000 и выше**. Файлы, созданные такими пользователями, обычно имеют **664** разрешения. Каталоги, созданные такими пользователями, обычно имеют **775** разрешений. Пользователи могут переопределить настройки по умолчанию, добавив команду **umask** в свои **~/.bashrc** или **~/.bash_profile**.

ЦЕЛЬ СЕРТИФИКАЦИИ 4.02

Списки контроля доступа ACL и многое другое

Было время, когда пользователи имели доступ для чтения к файлам всех других пользователей. Однако по умолчанию пользователи имеют разрешения только в своих собственных каталогах. С помощью **ACL** вы можете дать выбранным пользователям права на чтение, запись и выполнение для выбранных файлов в вашем домашнем каталоге. Это обеспечивает второй уровень дискреционного контроля доступа, метод, который поддерживает переопределение стандартных разрешений **ugo/rwx**.

Строго говоря, обычные разрешения **ugo/rwx** - это первый уровень дискреционного контроля доступа. Другими словами, **списки ACL** начинаются с владения и прав, описанных ранее в этой главе. Вскоре вы увидите, как это отображается с помощью **команд ACL**.

Чтобы настроить **ACL**, вам нужно смонтировать соответствующую файловую систему с **опцией acl**. Затем вам нужно настроить разрешения на выполнение для связанных каталогов. Только тогда вы можете настроить **ACL** с желаемыми разрешениями для соответствующих пользователей.

ACL поддерживаются в файловых системах **ext4** и **XFS**, а также в сетевой файловой системе (**NFS**) версии 4.

Команда getfacl

Предполагая, что **пакет acl** установлен, у вас должен быть доступ к команде **getfacl**, которая отображает текущие списки **ACL** файла. Например, следующая команда отображает текущие разрешения и **ACL** для файла **anaconda-ks.cfg** в каталоге **/root**:

```
[root@server1 ~]# getfacl anaconda-ks.cfg
# file: anaconda-ks.cfg
# owner: root
# group: root
user::rw-
group::---
other::---
```

Запустите команду **ls -l /root/anaconda-ks.cfg**. Вы должны распознавать каждый элемент выходных данных, показанных здесь: поскольку в файле **anaconda-ks.cfg** не заданы списки **ACL**, команда **getfacl** отображает только стандартные разрешения и владельца. **Списки ACL**, которые вы добавите в ближайшее время, выходят за рамки разрешений, указанных здесь. Но сначала вам может понадобиться сделать файловую систему **дружественной** к этому **второму уровню ACL**.

Сделать файловую систему доступной для ACL

RHEL 7 использует файловую систему **XFS**. Когда вы создаете файловую систему **XFS** или **ext2/ext3/ext4** в RHEL 7, **ACL** включаются по умолчанию. С другой стороны, файловые системы **ext2**, **ext3** и **ext4**, созданные в более старых версиях Red Hat, могут автоматически не включать поддержку ACL.

!!!! On the Job !!!!
Чтобы проверить, включена ли в файловой системе ext2/ext3/ext4 опция монтирования acl по умолчанию на устройстве с разделами, например, как /dev/sda1, запустите команду tune2fs -l /dev/sda1. Помните, что для файловых систем XFS и всех файловых систем ext, созданных в RHEL 7, поддержка ACL включена по умолчанию. Следовательно, монтирование файловой системы с опцией acl потребуется только в файловых системах ext, созданных в более старых версиях Red Hat Enterprise Linux или в файловых системах ext2/ext3/ext4, где опция acl была явно удалена.
!!!!!

Если вы хотите включить поддержку **ACL** в файловой системе, для которой не настроен параметр монтирования **acl**, вы можете соответствующим образом перемонтировать существующий раздел. Например, мы можем перемонтировать раздел **/home с помощью ACL**, используя следующую команду:

 # mount -o remount -o acl /home

Чтобы убедиться, что **/home** монтируется при следующей перезагрузке, отредактируйте **/etc/fstab**. На основании предыдущей команды связанная строка может выглядеть следующим образом, если **/home** отформатирован с помощью **ext4**:

/dev/sda3 /home ext4 defaults,acl 1,2

После внесения изменений в **/etc/fstab** вы можете активировать его с помощью следующей команды:

 # mount -o remount /home

Чтобы убедиться, что каталог **/home** смонтирован **с опцией acl**, выполните только одну команду **mount**, без ключей и параметров. Вы должны увидеть **acl** в выходных данных, как показано здесь:

/dev/sda3 on /home type ext4 (rw,acl)

Теперь вы можете начать работать с командами **ACL** для установки списков контроля доступа к нужным файлам и каталогам.

Управление списками ACL для файла

Теперь с правильно смонтированной файловой системой и соответствующими разрешениями вы можете управлять списками ACL в системе. Чтобы просмотреть текущие списки ACL, выполните команду имени файла **getfacl**. Для этого примера мы создали текстовый файл с именем **TheAnswers** в каталоге **/home/examprep**. Ниже приведен вывод команды **getfacl /home/examprep/TheAnswers**:

 # file home/examprep/TheAnswers
 # owner: examprep
 # group: proctors
user::rw-
group::r--
other::---

Обратите внимание, что файл **TheAnswers** принадлежит пользователю **examprep** и группе **proctors**. Этот владелец пользователя имеет права на чтение и запись; владелец этой группы имеет права на чтение этого файла. Другими словами, в то время как пользователь **examprep** может читать и изменять этот файл, пользователи-члены группы **proctors** могут его читать.

Теперь, если вы были пользователем **examprep** или пользователем **root** в этой системе, вы можете **назначить ACL для файла** с именем **TheAnswers** для меня (пользователь **michael**) с помощью команды **setfacl**. Например, следующая команда дает **michael** права на чтение, запись и выполнение для этого файла:

 # setfacl -m u:michael:rwx /home/examprep/TheAnswers

Эта команда изменяет **ACL** для указанного файла, изменяя **(-m**) **ACL** для пользователя **michael**, предоставляя этому пользователю права на чтение, запись и выполнение для этого файла. Для подтверждения выполните команду **getfacl** для этого файла, как показано на рисунке 4-2.

РИСУНОК 4-2 ACL файла

```
[root@server1 ~]# getfacl /home/examprep/TheAnswers
getfacl: Removing leading '/' from absolute path names
# file: home/examprep/TheAnswers
# owner: examprep
# group: examprep
user::rw-
user:michael:rwx
group::r--
mask::rwx
other::r--

[root@server1 ~]#
```

Но когда мы попытались получить доступ к этому файлу из учетной записи **michael**, это не сработало. На самом деле, если мы пытаемся получить доступ к файлу с помощью текстового редактора **vi**, он предполагает, что **/home/examprep/TheAnswers** - это новый файл. Затем он отказывается сохранять любые изменения, которые мы можем внести в этот файл.

Прежде чем файлы из **каталога /home/examprep** станут доступны, пользователю с правами администратора необходимо либо изменить разрешения, либо параметры ACL, связанные с этим каталогом. Прежде чем мы перейдем к изменению дискреционных элементов управления доступом в каталоге, давайте рассмотрим некоторые различные параметры команды **setfacl**.

Несмотря на имя, команда **setfacl** может использоваться для удаления таких привилегий **ACL** с **ключом -x**. Например, следующая команда удаляет ранее настроенные разрешения **rwx** для пользователя **michael**:

 # setfacl -x u:michael /home/examprep/TheAnswers

Кроме того, команда **setfacl** может использоваться с группами; например, если группа **teachers** существует, следующая команда предоставит права на чтение пользователям, которые являются членами этой группы:

 # setfacl -m g:teachers:r /home/examprep/TheAnswers

Вы также можете использовать команду **setfacl**, чтобы удалить все разрешения от имени пользователя. Например, следующая команда запрещает доступ к каталогу **/home/examprep** для пользователя **michael**:

 # setfacl -m u:michael:- /home/examprep

Если вы хотите посмотреть, как работают ACL, не удаляйте привилегии ACL в файле **TheAnswers**, по крайней мере пока. В качестве альтернативы, если вы хотите начать все сначала, следующая команда с **ключом -b** удалит все записи ACL в указанном файле:

 # setfacl -b /home/examprep/TheAnswers

Некоторые из ключей, доступных для команды **setfacl**, показаны в **таблице 4-5**.

ТАБЛИЦА 4-5 Описание прав доступа к файлу

Ключи	Описание
-b (--remove-all)	Удаляет все записи **ACL**; сохраняет стандартные разрешения **ugo/rwx**
-k	Удаляет записи **ACL** по умолчанию
-m	Изменяет **ACL** файла, обычно с определенным пользователем **(u)** или группой **(g)**
-n (--mask)	Пропускает пересчет записи маски
-R	Применяет изменения рекурсивно
-x	Удаляет конкретную запись **ACL**

Один немного опасный вариант относится к другим пользователям. Например, команда

 # setfacl -m o: rwx /home/examprep/TheAnswers

позволяет другим пользователям читать, записывать и выполнять разрешения для файла **TheAnswers**. Это происходит путем изменения основных прав доступа к файлу, как показано в выводе на **ls -l /home/examprep/TheAnswers**. Ключи **-b** и **-x** не удаляют такие изменения; вам нужно будет использовать следующую команду:

 # setfacl -m o:- /home/examprep/TheAnswers

Настройте каталог для ACL

Существует несколько способов настроить каталог для обмена файлами с ACL. Во-первых, вы можете установить обычный бит выполнения для всех остальных пользователей. Один из способов сделать это в указанном каталоге - выполнить следующую команду:

 # chmod 701 /home/examprep

Это минимальный способ предоставления доступа к файлам в каталоге. Пользователи, кроме **examprep** и **root**, не могут перечислить файлы в этом каталоге. Они должны знать, что файл **TheAnswers** действительно существует для доступа к этому файлу.

Однако с битом выполнения, установленным для других пользователей, любой пользователь может получить доступ к файлам в каталоге **/home/examprep**, для которого у нее есть разрешение. Это должно поднять флаг безопасности. Любой пользователь? Несмотря на то, что файл скрыт, хотите ли вы когда-нибудь дать реальные привилегии всем пользователям? Конечно, ACL были установлены только для файла **TheAnswers** в этом каталоге **/home/examprep**, но это один уровень безопасности, который вы добровольно убрали.

Правильный подход - применить команду **setfacl** к каталогу **/home/examprep**. Самый безопасный способ настроить совместное использование - установить разрешения на выполнение **ACL** только для учетной записи **michael** пользователя в указанном каталоге с помощью следующей команды:

 # setfacl -m u:michael:x /home/examprep

Поскольку пользователь **examprep** является владельцем каталога **/home/examprep**, этот пользователь также может выполнить указанную команду **setfacl**.

Иногда вы можете захотеть применить такие **ACL** ко всем файлам в каталоге. В этом случае **ключ -R** может использоваться для рекурсивного применения изменений; например, следующая команда позволяет пользователю **michael** иметь разрешения на чтение и выполнение для всех файлов в каталоге **/home/examprep**, а также для любых подкаталогов, которые могут существовать:

 # setfacl -R -m u:michael:rx /home/examprep

Есть два метода, доступных для отмены этих опций. Во-первых, вы можете применить **ключ -x** к предыдущей команде, пропустив настройки разрешений:

 # setfacl -R -x u:michael /home/examprep

В качестве альтернативы вы можете использовать **ключ -b**; однако это приведет к удалению **списков ACL**, настроенных для всех пользователей в указанном каталоге (и с **ключом -R**, в соответствующих подкаталогах):

 # setfacl -R -b /home/examprep

Настройте списки ACL по умолчанию

Каталоги также могут содержать один или несколько списков ACL по умолчанию. Концепция ACL по умолчанию аналогична обычной записи ACL, с той разницей, что ACL по умолчанию не влияет на текущие разрешения каталога, но наследуется файлами, созданными в каталоге.

Например, если вы хотите, чтобы все новые файлы и каталоги в **/home/examprep наследовали ACL**, который предоставляет права на чтение и выполнение пользователю **michael**, вы можете выполнить следующую команду:

 # setfacl -d -m u:michael:rx /home/examprep

Параметр **-d** в предыдущей команде указывает, что текущая операция применяется к **списку ACL** по умолчанию. Команда **getfacl** может отображать стандартные и списки по умолчанию ACL в указанном каталоге:

 # getfacl /home/examprep
getfacl: Removing leading '/' from absolute path names
 # file: home/examprep

 # owner: examprep
 # group: examprep
user::rwx
user:michael:--x
group::---
mask::--x
other::---
default:user::rwx
default:user:michael:r-x
default:group::---
default:mask::r-x
default:other::---

ACL и Маски

Маска, связанная с ACL, ограничивает разрешения, доступные для файла для именованных пользователей и групп, а также для владельца группы. Маска, показанная на **рисунке 4-2**, имеет вид **rwx**, что означает отсутствие ограничений. Если для него установлено значение **r**, то считываются только те права, которые могут быть предоставлены с помощью такой команды, как **setfacl**. Чтобы изменить маску в файле **TheAnswers** только для чтения, выполните следующую команду:

 # setfacl -m mask:r-- /home/examprep/TheAnswers

Теперь просмотрите результат с помощью команды **getfacl /home/examprep/TheAnswers**. Обратите внимание на запись для конкретного пользователя. Основываясь на **привилегиях ACL**, предоставленных пользователю **michael** ранее, вы увидите разницу с **рисунком 4-2**:

user:michael:rwx #effective:r--

Другими словами, с помощью маски **r--** вы можете попытаться предоставить другим пользователям все привилегии в мире. Но все, что можно установить с помощью этой маски, - это права на чтение.

!!!!! On the Job !!!
Маска влияет только на владельца группы и на именованных пользователей и группы. Это не влияет на пользователя-владельца файла и «другую» группу разрешений.
!!!!!

УПРАЖНЕНИЕ 4-1

Используйте ACL для запрета пользователя

В этом упражнении вы создадите **списки ACL**, чтобы запретить доступ к файлу конфигурации обратной связи обычному пользователю. Это файл **ifcfg-lo** в каталоге **/etc/sysconfig/network-scripts**. В этом упражнении предполагается, что вы настроили обычного пользователя. Поскольку в наших системах мы настроили пользователя **michael**, это обычный пользователь, указанный в этом упражнении. Замените соответственно на вашего пользователя. Чтобы запретить такой доступ, выполните следующие действия:

1. Создайте резервную копию текущего файла конфигурации для устройства обратной связи. Это файл **ifcfg-lo** в каталоге **/etc/sysconfig/network-scripts**. (Подсказка: используйте команду **cp**, а не команду **mv**.)
2. Выполните команду **setfacl -m u:michael:- /etc/sysconfig/network-scripts/ifcfg-lo**.
3. Просмотрите результаты. Запустите команду **getfacl** для обеих копий файла, в **/etc/sysconfig/network-scripts** и в каталоге резервного копирования. В чем различия?
4. Войдите как целевой пользователь. Из учетной записи администратора **root** один из способов сделать это - использовать команду **su - michael**.
5. Попробуйте прочитать файл **/etc/sysconfig/network-scripts/ifcfg-lo** в текстовом редакторе **vi** или даже с помощью команды **cat**. Что происходит?
6. Повторите предыдущий шаг с файлом в каталоге резервного копирования. Что происходит?
7. Теперь запустите команду **cp** из резервной копии файла **ifcfg-lo** и перезапишите текущую версию в файле **/etc/sysconfig/network-scripts**. (Не используйте команду **mv** для этой цели.) Для этого вам необходимо вернуться от имени пользователя **root**.
8. Попробуйте снова выполнить команду **getfacl /etc/sysconfig/network-scripts/ifcfg-lo**. Вы удивлены результатом?
9. Существует два способа **восстановить исходную конфигурацию ACL** для файла **ifcfg-lo**. Сначала примените к файлу команду **setfacl -b**. Это сработало? Подтвердите с помощью команды **getfacl**. Если были применены какие-либо другие связанные команды, это может сработать или не сработать.
10. Другой способ **восстановить исходный ACL**-файл - восстановить резервную копию, сначала удалив измененный файл в каталоге **/etc/sysconfig/network-scripts**, а затем скопировав файл из каталога резервных копий.
11. Однако, если вы выполните Шаг 10, вам также может понадобиться восстановить контексты **SELinux** файла с помощью следующей команды:

    restorecon -F /etc/sysconfig/network-scripts/ifcfg-lo

    Дополнительная информация о команде **restorecon** доступна далее в этой главе.

ACL для NFS

Хотя нет никаких доказательств того, что экзамены Red Hat охватывают списки ACL на основе NFS, это особенность, которую должны знать администраторы Linux. Таким образом, описание в этом разделе просто предоставляет примеры и далеко не полное. Для получения дополнительной информации обратитесь к справочной странице **nfs4_acl**, которая устанавливается RPM-пакетом **nfs4-acl-tools**.

Часто каталог **/home** берется из общего тома **NFS**. На самом деле, списки **ACL** на основе **NFS** более детализированы, чем **стандартные ACL**. Эта функция была представлена в NFS версии 4, стандарте RHEL 7. Для этого команда **nfs4_getfacl** может отображать списки **ACL**, связанные с файлами, в общем каталоге. На основании ранее предоставленных списков ACL на **рис. 4-3** показаны выходные данные команды **nfs4_getfacl.**

Вывод в формате

type:flags:principal:permissions

где настройки обозначены двоеточием. Вкратце, показанные два типа **разрешают** (**A**) или **запрещают** (**D**) указанному участнику (пользователю или группе) указанные разрешения. На **рисунке 4-3** не показаны флаги, которые могут обеспечить относительно детальный контроль. Принципал может быть обычным пользователем или группой в нижнем

регистре. Это также может быть общий пользователь, такой как файл **OWNER**, **GROUP**, которому принадлежит файл, или другие пользователи, как указано в **EVERYONE**. Разрешения, как **показано в Таблице 4-6**, позволяют очень детализированный контроль. Эффект зависит от того, является ли объект файлом или каталогом.

Настройка **NFS** в качестве клиента описана в **главе 6** с другими локальными и сетевыми файловыми системами. Конфигурация сервера **NFS** является целью RHCE, описанной в **главе 16**.

РИСУНОК 4-3 ACL NFS версии 4

```
[michael@server1 ~]$ nfs4_getfacl /test/examprep/
A::OWNER@:rwaDxtTcCy
A::michael@localdomain:xtcy
A::GROUP@:tcy
A::EVERYONE@:tcy
[michael@server1 ~]$ nfs4_getfacl /test/examprep/TheAnswers
D::OWNER@:x
A::OWNER@:rwatTcCy
A::michael@localdomain:rwaxtcy
A::GROUP@:rtcy
A::EVERYONE@:rtcy
[michael@server1 ~]$
```

ТАБЛИЦА 4-6 Описания разрешений ACL для NFSv4

Разрешение	Описание
r	Прочитать файл или список каталогов
w	Записать в файл или создать новый файл в каталоге
a	Добавить данные в файл или создать подкаталог
x	Выполнить программу или изменить каталог
d	Удалить файл или каталог
D	Удалить подкаталог
t	Прочитайте атрибуты файла или каталога
T	Напишите атрибуты файла или каталога
c	Прочитайте ACL файла или каталога
C	Запишите ACL файла или каталога
y	Разрешить клиентам использовать синхронный ввод/вывод для файла или каталога

ЦЕЛЬ СЕРТИФИКАЦИИ 4.03
Основное управление брандмауэром

Традиционно межсетевые экраны настраивались только между локальными сетями и внешними сетями, такими как Интернет. Но с ростом угроз безопасности растет потребность в брандмауэрах в каждой системе. RHEL 7 включает брандмауэры по умолчанию в любой конфигурации.

Ядро Linux поставляется с мощной структурой, **системой Netfilter**, которая позволяет другим модулям ядра предлагать такие функции, как **фильтрация пакетов, преобразование сетевых адресов (NAT) и распределение нагрузки**. Команда **iptables** является основным инструментом, который взаимодействует с системой **Netfilter** для обеспечения фильтрации пакетов и NAT.

Перед отправкой сообщения по IP-сети сообщение разбивается на более мелкие блоки, называемые пакетами. Административная информация, **включая тип данных, адрес источника и адрес назначения, добавляется к каждому пакету**. Пакеты повторно собираются, когда достигают конечного компьютера. Правило **iptables проверяет** эти административные поля в каждом пакете, чтобы определить, следует ли разрешить прохождение пакета.

!!!! Exam watch !!!!
RHEL 7 также включает команду брандмауэра для сетей IPv6, ip6tables. Связанные команды практически идентичны. В отличие от iptables, команда ip6tables не указана в задачах Red Hat.
!!!!!

Инструмент **iptables** является основной, которая используется другими службами для управления правилами брандмауэра системы. RHEL 7 поставляется с двумя такими сервисами: **новым демоном firewalld** и **сервисом iptables**, который был включен в предыдущие выпуски Red Hat Enterprise Linux. Вы можете взаимодействовать с **firewalld** с помощью графической утилиты **firewall-config** или клиентом командной строки **firewall-cmd.**

Службы **iptables** и **firewalld** используют **систему Netfilter** в ядре Linux для фильтрации пакетов. Однако, хотя **iptables** основан на концепции «цепочки правил фильтрации (**chain of filter rules**)» для блокировки или пересылки трафика, **firewalld** «основан на зонах (**zone-based**)», как вы увидите в следующих разделах.
Существуют требования RHCSA и RHCE, связанные с настройкой и управлением брандмауэром. Для RHCSA вам необходимо понять, как настроить брандмауэр для блокировки или разрешения сетевой связи через один или несколько портов с использованием **iptables**, **firewall-config** или **firewall-cmd**. Для RHCE вам необходимы более глубокие знания **firewalld** и его функций, таких как «расширенные правила, зоны и пользовательские правила, для реализации фильтрации пакетов и настройки преобразования сетевых адресов (NAT) **(rich rules, zones and custom rules, to implement packet filtering and configure network address translation (NAT))** ».

ТАБЛИЦА 4-7 Общий TCP/IP Порты

Порт	Описание
20, 21	**FTP**
22	Безопасная оболочка (**SSH**)
23	**Telnet**
25	Простой протокол пересылки почты (**SMTP**); например, **Postfix, sendmail**
53	Серверы службы доменных имен
80	Протокол передачи гипертекста (**HTTP**)
88	**Kerberos**
110	Протокол почтового отделения, версия 3 (**POP3**)
139	Сетевая служба сеансов базовой системы ввода/вывода (**NetBIOS**)
143	Протокол доступа к интернет почте (**IMAP**)
443	**HTTP**, безопасный (**HTTPS**)

Стандартные порты

Linux связывается по сети, в основном используя **набор протоколов TCP/IP**. Различные службы по умолчанию используют определенные порты и протоколы, как определено в файле **/etc/services**. Может быть полезно знать некоторые из этих портов наизусть, например, те, которые описаны **в Таблице 4-7**. Помните, что некоторые из этих портов могут обмениваться данными с использованием протокола управления передачей (**TCP**), протокола пользовательских дейтаграмм (**UDP**) или даже протокола передачи управления потоком (**SCTP**). Например, как отмечено в следующих выдержках из файла **/etc/services**, службе **FTP** назначены перечисленные здесь порты **TCP** и **UDP**:

ftp-data 20/tcp
ftp-data 20/udp

ftp 21/tcp
ftp 21/udp

Однако вскоре вы увидите, что инструменты настройки брандмауэра Red Hat открывают только соединения **TCP** для служб **FTP**, и сервер **vsFTP** по умолчанию, настроенный в **главе 1**, работает в таких условиях нормально. Это связано с тем, что стандартная политика Управления по назначению номеров в Интернете (IANA) заключается в регистрации номеров портов для **TCP** и **UDP**, даже если служба поддерживает только протокол **TCP**.

Фокус на iptables

Философия, лежащая в основе **iptables**, основана на «цепочках (**chains**)». Это наборы правил, применяемые к каждому сетевому пакету, объединенные вместе. Каждое правило выполняет две вещи: оно определяет условия, которым должен соответствовать пакет, чтобы соответствовать правилу, и определяет действие, если пакет соответствует.

Команда **iptables** использует следующий базовый формат:

iptables -t tabletype <action_direction> <packet_pattern> -j <what_to_do>

Теперь давайте проанализируем эту команду, шаг за шагом. Первый - это ключ **-t tabletype.** Для **iptables** есть два основных параметра типа таблицы:

- **filter** Устанавливает правило для фильтрации пакетов.
- **nat** Конфигурирует трансляцию сетевых адресов, также известную как **маскарадинг**, которая обсуждается далее **в главе 10**.

По умолчанию это **filter**; если вы не указали **-t tabletype**, команда **iptables** предполагает, что она применяется, как правило к фильтрации пакетов.

Следующим является <**action_direction**>. Четыре основных действия связаны с правилами **iptables**:

- **-A (--append)** Добавляет правило в конец цепочки.
- **-D (--delete)** Удаляет правило из цепочки. Укажите правило по номеру или шаблону пакета.
- **-L (--list)** Список текущих настроенных правил в цепочке.
- **-F (--flush)** Сбрасывает все правила в текущей цепочке **iptables**.

Если вы **добавляете (-A)** или **удаляете из (-D) цепочки**, вы хотите применить ее к сетевым данным, перемещающимся в одном из трех направлений:

- **INPUT** Все входящие пакеты проверяются на соответствие правилам в этой цепочке.
- **OUTPUT** Все исходящие пакеты проверяются на соответствие правилам в этой цепочке.
- **FORWARD** Все пакеты, полученные с компьютера и отправленные на другой компьютер, проверяются на соответствие правилам в этой цепочке. Другими словами, это пакеты, которые маршрутизируются через локальный сервер.

Как правило, каждое из этих направлений является названием цепочки.

Далее вам необходимо настроить <**packet_pattern**>. Все брандмауэры **iptables** проверяют каждый пакет на соответствие этому шаблону. **Самый простой шаблон по IP-адресу:**

- **-s ip_address** Все пакеты проверяются на определенный **IP-адрес источника**.
- **-d ip_address** Все пакеты проверяются на определенный **IP-адрес назначения**.

Шаблоны пакетов могут быть более сложными. В **TCP/IP пакеты** транспортируются с использованием **протокола TCP, UDP** или **ICMP**. Вы можете указать протокол с **ключом -p**, за которым следует порт назначения **(--dport).** Например, расширение **-p tcp --dport 80** влияет на пользователей вне вашей сети, которые пытаются установить **HTTP-соединение**.

Как только команда **iptables** находит совпадение **с шаблоном пакета**, ей нужно знать, **что делать с этим пакетом**, что приводит к последней части **команды -j <what_to_do**>. Есть три основных варианта:

- **DROP** Пакет отброшен. На запрашивающий компьютер сообщения не отправляются.
- **REJECT** Пакет отброшен. Сообщение об ошибке отправляется запрашивающему компьютеру.
- **ACCEPT** Пакет может следовать, как указано в действии **-A: INPUT, OUTPUT** или **FORWARD**.

Посмотрите на некоторые примеры того, как вы можете использовать команды **iptables** для настройки брандмауэра. Первым делом всегда нужно посмотреть, что в данный момент настроено, с помощью следующей команды:

 # iptables -L

Если брандмауэр **iptables** настроен, он должен возвращать правила цепочки как минимум в трех различных категориях: **INPUT, FORWARD** и **OUTPUT.**

Держите брандмауэр в рабочем состоянии

Брандмауэры Linux, такие как **firewalld** и служба **iptables**, основаны на команде **iptables**. Чтобы просмотреть текущие правила, выполните команду **iptables -L**. Предположим, все, что вы видите, это следующий пустой список правил:

Chain INPUT (policy ACCEPT)
target prot opt source destination

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination

Этот вывод означает, что служба **firewalld** может быть не включена. В RHEL 7 **firewalld** является службой межсетевого экрана по умолчанию. Убедитесь, что он работает:

 # systemctl status firewalld

Если служба не активна, убедитесь, что служба **iptables** отключена, затем запустите **firewalld** и убедитесь, что она включена при загрузке:

 # systemctl stop iptables
 # systemctl disable iptables

 # systemctl start firewalld
 # systemctl enable firewalld

Прежде чем перейти к настройке **firewalld**, мы кратко рассмотрим сервис iptables. Помимо того, что он является обязательным требованием для сдачи экзамена RHCSA, **базовые знания службы iptables** обеспечат лучшее понимание более продвинутых функций, предоставляемых **firewalld**.

Сервис iptables

В то время как служба **iptables** была брандмауэром по умолчанию, работающим в RHEL 6, **firewalld** является настройкой по умолчанию в RHEL 7. При желании вы можете отключить **firewalld** в RHEL 7 и **переключиться** на старую службу **iptables**. Для этого выполните следующие команды:

 # systemctl stop firewalld
 # systemctl disable firewalld
 # systemctl start iptables
 # systemctl enable iptables

Аналогично, чтобы вернуться к **firewalld**, выполните команды, перечисленные в предыдущем разделе. После запуска службы **iptables** просмотрите существующие правила брандмауэра с помощью **iptables -L**. Вывод системы по умолчанию **server1.example.com** показан на **рисунке 4-4**.

На **рисунке 4-4** показаны шесть столбцов информации, которые соответствуют различным параметрам команды **iptables**. Показанный брандмауэр основан на следующих правилах, перечисленных в файле **/etc/ ysconfig/iptables**. Первая строка в файле указывает, что правила, которым нужно следовать, являются правилами фильтрации. Альтернативные правила поддерживают преобразование сетевых адресов (**NAT**) или манипуляции различными параметрами пакетов в таблице **mangle**.

***filter**

РИСУНОК 4-4 Правила брандмауэра для службы iptables

```
[root@server1 ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@server1 ~]#
```

Затем сетевой трафик, направляемый в локальную систему, предназначенный для пересылки и отправки, обычно принимается по умолчанию с опцией **ACCEPT**. Часть **[0: 0]** показывает количество байтов и пакетов.

:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]

Следующие строки применяются к команде **iptables**. Все **ключи** и **опции**, перечисленные в этом файле, должны быть доступны на соответствующей странице руководства (**man**).

Следующая строка поддерживает текущую сетевую связь. Опция **ESTABLISHED** продолжает принимать входящие пакеты, связанные с входящими сетевыми подключениями. Опция **RELATED** принимает пакеты для последующих сетевых подключений, например для передачи данных по **FTP**.

-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

Следующая строка принимает пакеты, **связанные с ICMP**, чаще всего связанные с командой **ping**. Когда пакет отклонен, соответствующее сообщение также использует протокол **ICMP**.

-A INPUT -p icmp -j ACCEPT

Следующая строка добавляет **add (-A)** правило в цепочку **INPUT**, связанное с сетевым интерфейсом **(-i),** известным, как **адаптер обратной связи (lo).** Любые данные, обработанные этим устройством, переходят **jumps (-j)** к принятию (**acceptance**).

-A INPUT -i lo -j ACCEPT

Следующая строка - единственная, которая напрямую принимает новые обычные сетевые данные, используя протокол TCP, через все интерфейсы. Правило ищет совпадение **match (-m)** для НОВОГО (**NEW**) состояния соединения **(--state NEW**), для **сопоставления пакетов TCP**, используя **протокол TCP (-p tcp),** отправленный на порт **назначения (--dport) 22**, который **соответствует услуге SSH**. Сетевые пакеты, которые соответствуют всем этим критериям, принимаются **accepted** (-j **ACCEPT**). Как только соединение установлено, первое обычное правило, описанное в этой главе, продолжает принимать пакеты по этому установленному соединению.

-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT

Последние два правила отклоняют все остальные пакеты, и отправляющее сообщение в исходную систему отправляет сообщение о запрете **icmp-host-prohibited**:

-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited

COMMIT завершает список правил:

COMMIT

Поскольку этот раздел связан с экзаменом RHCSA, более подробное обсуждение можно найти в **главе 10**. На этом уровне вам необходимо знать, как управлять этими межсетевыми экранами с помощью стандартных инструментов настройки.

Служба firewalld

Вы можете автоматизировать процесс настройки брандмауэра. Для этого в RHEL 7 **firewalld** поставляется как управление с консоли, так и с инструментом настройки графического интерфейса. Хотя внешний вид этих двух приложений различен, вы можете использовать оба инструмента для настройки доступа к доверенным службам. Перед запуском средства настройки **firewalld** ознакомьтесь с шагами в предыдущем разделе «Поддержите этот брандмауэр в работе **(Keep That Firewall in Operation)**», чтобы убедиться, что **firewalld** работает и автоматически запускается во время процесса загрузки.

Служба **firewalld** предлагает те же функции, что и инструмент **iptables**, и многое другое. Одной из новых функций **firewalld** является межсетевой экран на основе зон. В брандмауэре на основе зон (***zone-based***) сети и интерфейсы группируются в зоны, причем каждая зона настроена с различным уровнем доверия. Зоны, определенные в **firewalld**, перечислены в **таблице 4-8** вместе с их поведением по умолчанию для исходящих и входящих соединений.

!!!!! On the Job !!!!!
Зона состоит из группы сетевых адресов и интерфейсов источника, а также правил обработки пакетов, соответствующих этим адресам источника и сетевым интерфейсам.
!!!!

ТАБЛИЦА 4-8 Зоны в огненном мире

Зона	Исходящие соединения	Входящие подключения
drop	Разрешается	Сброшено.
block	Разрешается	Отклонено с сообщением, запрещенным icmp-host.
public	Разрешается	DHCPv6 клиент и SSH разрешены.
external	Разрешено и маскируется под IP-адрес исходящего сетевого интерфейса	SSH разрешен
dmz	Разрешается	SSH разрешен
work	Разрешается	Клиент DHCPv6, IPP и SSH разрешены.
home	Разрешается	Клиент DHCPv6, многоадресный DNS, IPP, клиент Samba и SSH разрешены.
internal	Разрешается	То же, что и домашняя зона.
trusted	Разрешается	Разрешается.

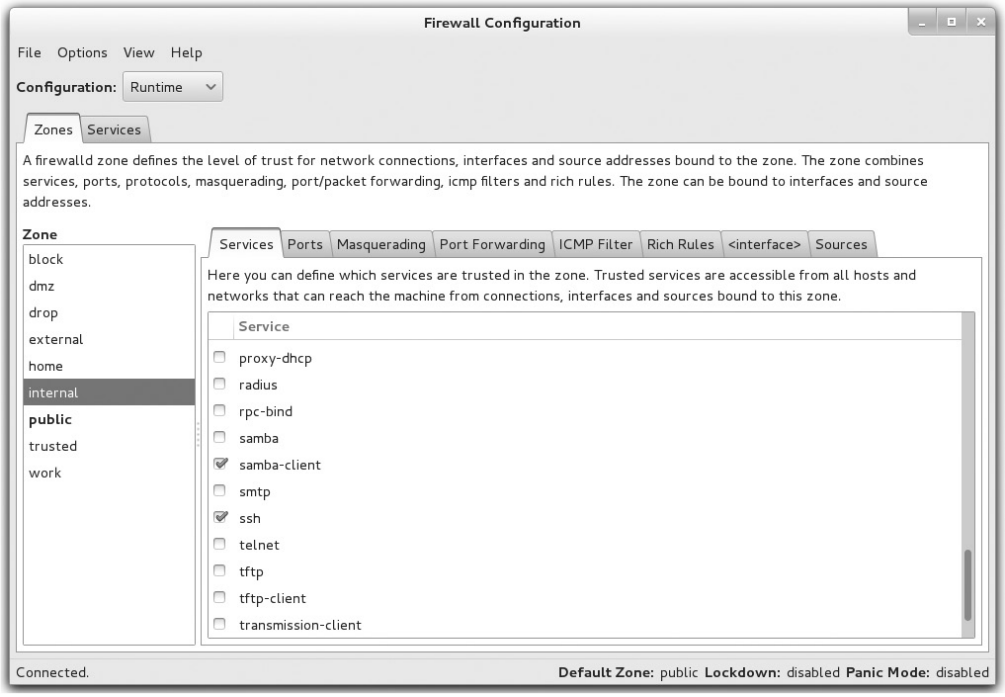
Инструмент настройки межсетевого экрана с графическим интерфейсом

Вы можете запустить графический инструмент настройки **firewalld** из командной строки на основе графического интерфейса с помощью команды **firewall-config**. Либо в среде рабочего стола **GNOME** выберите **Приложения | Разное | Брандмауэр (Applications | Sundry | Firewall)**. Результат показан на **рисунке 4-5**.

Как показано на рисунке, главное окно содержит различные меню и вкладки. В верхнем левом углу есть раскрывающееся меню «**Конфигурация**», в котором вы можете установить брандмауэр в режим **Runtime** или **Permanent**. Если для него установлено время выполнения (**Runtime**), изменения, примененные с помощью **firewall-config**, вступают в силу немедленно, но не сохраняются после перезагрузки сервера. Также можно выбрать постоянный режим **(Permanent)** , чтобы изменения вступили в силу после перезагрузки сервера. В любое время вы можете нажать **Параметры | Перезагрузите**

Firewalld (**Options | Reload Firewalld**), чтобы новая конфигурация брандмауэра сразу вступила в силу.

РИСУНОК 4-5 Графический инструмент настройки брандмауэра



!!!!! On the Job !!!!
Вы можете изменять определения зон и служб только в постоянном режиме.
!!!!!

Вкладка «**Зона**»(**Zone**) включает в себя все зоны, ранее перечисленные в **таблице 4-8**. Когда входящий пакет попадает в межсетевой экран, его исходный адрес проверяется на соответствие сетевым адресам, которые принадлежат существующим зонам. Если совпадений не найдено, входящий интерфейс пакета проверяется, чтобы проверить, принадлежит ли он к зоне. Как только соответствие найдено, пакет обрабатывается в соответствии с правилами зоны, которой он соответствует.

В главном окне настройки брандмауэра публичная (**public)** зона отображается жирным шрифтом, чтобы указать, что эта зона является зоной по умолчанию. Зона по умолчанию имеет особое значение: любой новый сетевой интерфейс, добавленный в систему, автоматически назначается зоне по умолчанию. Кроме того, правила зоны по умолчанию обрабатываются для всех входящих пакетов, которые не соответствуют ни одной из других зон. Вы можете установить другую зону по умолчанию, щелкнув **Параметры | Изменить зону по умолчанию(Options | Change Default Zone)**.

Чтобы разрешить или запретить входящий трафик через брандмауэр, выберите зону и добавьте или снимите флажок на вкладке «**Службы**»(**Services**) для службы, которую вы хотите предоставить или заблокировать. В качестве альтернативы вы также можете указать протокол и порт на вкладке «**Порты**»(**Ports**).

В **firewalld** служба определяется как группа из одного или нескольких протоколов и портов. Служба также может включать вспомогательный модуль **Netfilter** для поддержки фильтрации для тех приложений, которые динамически открывают несколько соединений.

Различные сетевые службы уже определены в окне **«Службы»** (**Services**). Наиболее распространенные описаны в **таблице 4-9**.

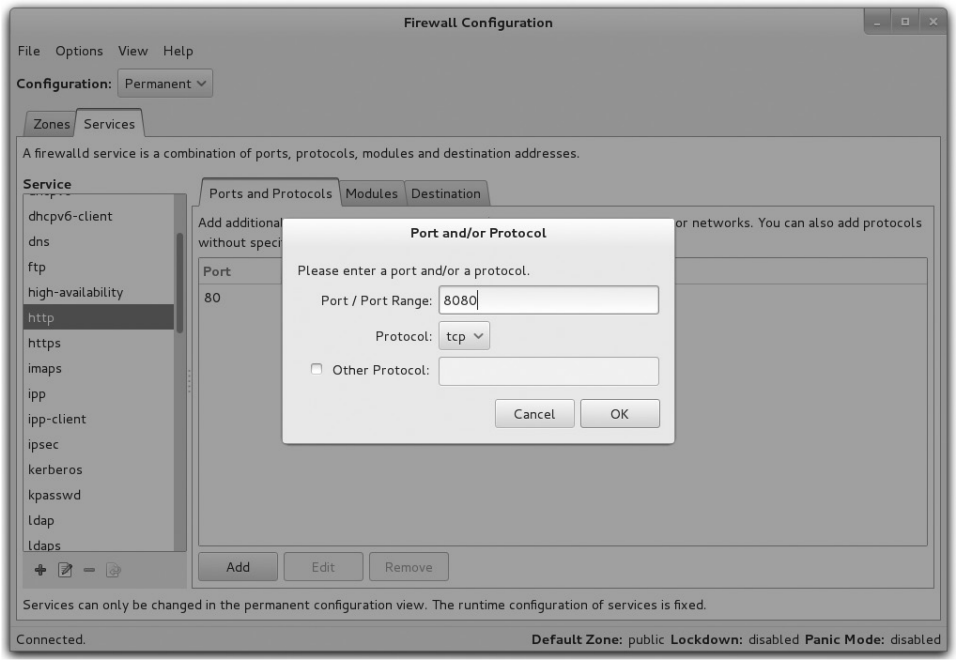
ТАБЛИЦА 4-9. Общие порты TCP/IP

Сервисы	Описание
amanda-client	Клиент Advanced Maryland Automatic Network Disk Archiver (AMANDA), связанный с UDP и TCP-портом 10080.
bacula	Сетевой резервный сервер с открытым исходным кодом; связан с портами TCP 9101, 9102 и 9103.
bacula-client	Клиент для сервера Bacula; связан с портом TCP 9102.
dhcp	Протокол динамической конфигурации хоста (DHCP) связан с портом UDP 67.
dhcpv6-client	Клиент DHCP на IPv6 связан с портом UDP 546.
dns	Сервер службы доменных имен (DNS); связан с портом 53 с использованием протоколов TCP и UDP.
ftp	Сервер протокола передачи файлов (FTP), связанный с TCP-портом 21; вспомогательный модуль Netfilter отслеживает динамические соединения, установленные для передачи данных по FTP.
http	Известный веб-сервер использует TCP-порт 80.
https	Для связи с защищенным веб-сервером через протокол SSL используется порт TCP 443.
imaps	IMAP через SSL обычно использует TCP-порт 993.
ipsec	Связан с UDP-портом 500 для ассоциации безопасности Интернета и протокола управления ключами (ISAKMP) вместе с протоколами транспортного уровня ESP и AH.
mdns	Многоадресный DNS (mDNS) связан с портом UDP 5353 и с многоадресным IP-адресом 224.0.0.251; mDNS часто используется для поддержки Linux реализации сетей с нулевой конфигурацией (zeroconf), известной как Avahi.
nfs	NFS версии 4 использует TCP-порт 2049.
ipp	Стандартный клиент сетевого сервера печати использует порты TCP и UDP 631, основанные на Интернет-протоколе печати (IPP).
ipp-client	Стандартный сетевой клиент печати использует UDP-порт 631, основанный на IPP-протоколе.
openvpn	Виртуальная частная сеть с открытым исходным кодом, которая использует UDP-порт 1194.
pop3s	POP-3 через SSL (Secure Sockets Layer) обычно использует TCP-порт 995.
radius	Протокол удаленной аутентификации (RADIUS) использует порты UDP 1812 и 1813.
samba	Протокол Linux для связи в сетях Microsoft использует TCP-порты 139 и 445, а также UDP-порты 137 и 138.
samba-client	Протокол Linux для взаимодействия с клиентами в сетях Microsoft использует UDP-порты 137 и 138.
ssh	Сервер SSH использует TCP-порт 22.
smtp	Сервер Simple Mail Transport Protocol, такой как sendmail или Postfix, использует TCP-порт 25.
tftp	Для связи с сервером Trivial File Transfer Protocol (TFTP) требуется UDP-порт 69.
tftp-client	Клиент TFTP использует динамический диапазон портов для передачи данных; вспомогательный модуль Netfilter отслеживает эти соединения.

Если вы переключите инструмент **firewall-config** в постоянный (**Permanent**) режим, вы можете добавить новые сервисы или отредактировать существующие. Для выполнения этой задачи выделите нижнюю часть окна «**Службы**»(**Services)** и щелкните соответствующий значок, чтобы удалить, добавить или изменить службу. При желании вы

также можете настроить пользовательские порты для существующей службы, щелкнув значок «Добавить» или «Изменить», как показано на **рис. 4-6**.

РИСУНОК 4-6. Добавление пользовательских портов в службу в инструменте firewall-config



Конфигурационный инструмент firewall-cmd в консоли

Средство настройки **firewall-cmd** имеет те же функции и службы, что и соответствующее средство графического интерфейса. Фактически, и графический инструмент настройки межсетевого экрана, и командный интерфейс **firewall-cmd** - это просто клиентские интерфейсы, которые взаимодействуют с нижележащим демоном **firewalld**.

Как и в случае с инструментом с графическим интерфейсом, **firewall-cmd** может отображать все доступные зоны и переключаться на другую зону по умолчанию. В следующем примере зона по умолчанию изменяется с общедоступной на внутреннюю зону:

```
# firewall-cmd --get-default-zone
public
# firewall-cmd --set-default-zone=internal
success
# firewall-cmd --get-default-zone
internal
#
```

Опция **--list-all** особенно полезна. В нем перечислены все настроенные интерфейсы и службы, разрешенные для зоны, как показано ниже:

```
# firewall-cmd --list-all
internal (default, active)
interfaces: eth0
sources:
services: dhcpv6-client ipp-client mdns samba-client ssh
ports:
```

masquerade: no
forward-ports:
icmp-blocks:
rich rules:
#

Как и во многих параметрах команды **firewall-cmd**, зона по умолчанию предполагается, если зона не указана с помощью ключа команды **--zone**. Вы можете добавлять и удалять порты и сервисы из зоны с помощью **--add-port, --add-service, --remove-port** и **--remove-service** - соответственно. В следующем примере включается служба **http** для трафика, попадающего в зону **dmz**:

 # firewall-cmd --zone=dmz --add-service=http
success
#

По умолчанию все изменения конфигурации, сделанные **firewall-cmd**, не сохраняются после перезагрузки сервера. Чтобы внести изменения, которые сохранятся после перезагрузки, добавьте параметр **--permanent** в **firewall-cmd**. Затем запустите **firewall-cmd --reload**, чтобы немедленно внести изменения.

!!!! Exam watch !!!!
Вы хотите изменения брандмауэра, которые выживают после перезагрузки. Для этого с помощью команды firewall-cmd используйте ключ --permanent.
!!!!

УПРАЖНЕНИЕ 4-2
Настройте параметры брандмауэра

В этом упражнении вы настроите брандмауэры из интерфейса командной строки и просмотрите результаты с помощью команд **nmap** и **telnet**. Хотя неважно, как вы решаете проблему на экзамене Red Hat, в этом упражнении вы увидите, что происходит при добавлении новой службы с помощью инструмента **firewall-cmd**. Конечно, можно использовать графический инструмент **firewall-config** для выполнения тех же задач. Предполагается, что система со стандартной конфигурацией **firewalld** описана в этой главе.

1. Просмотрите текущие активные службы в локальной системе с помощью команды **nmap localhost**. Определите **IP-адрес** локальной системы с помощью **команды ip addr**. Если локальной системой является **server1.example.com**, этот IP-адрес **должен быть 192.168.122.50**.
2. Убедитесь, что **firewalld** в данный момент работает выполните команду **systemctl status firewalld**.
3. Перейдите в другую систему. Вы можете сделать это в другой виртуальной машине или получить удаленный доступ к ней с помощью команды **ssh**. Если система **tester1.example.com** работает, вы можете войти в систему с помощью команды **ssh 192.168.122.150**.
4. Используйте команду **nmap** для просмотра того, что видно через брандмауэр; для отмеченной системы **server1.example.com** правильная команда будет **nmap 192.168.122.50**. Если IP-адрес, найденный на шаге 1, отличается, замените его соответствующим образом.
5. Вернитесь к исходной системе. Выполните следующие команды, чтобы установить и запустить службу **telnet**:

 # yum install telnet-server
 # systemctl start telnet.socket

6. Выполните следующую команду, чтобы показать текущие настройки для зоны по умолчанию:

   # firewall-cmd --list-all

7. Разрешить трафик **telnet** через зону по умолчанию. Не забудьте **ключ --permanent**, чтобы сделать изменения постоянными:

   # firewall-cmd --permanent --add-service=telnet

8. Примените предыдущее изменение конфигурации для немедленного исполнения, перегрузив брандмауэр:

   # firewall-cmd –reload

9. Вернитесь к системе **tester1.example.com**, как это было сделано в шаге 3.
10. Повторите шаг 4. Что вы видите?

ЦЕЛЬ СЕРТИФИКАЦИИ 4.04

Защита SSH с помощью аутентификации на основе ключей

Глава 2 посвящена **клиентским программам SSH**, включая **ssh, scp** и **sftp**. В этом разделе основное внимание уделяется обеспечению **доступа SSH с помощью аутентификации на основе ключей**.

Поскольку **SSH** является важным инструментом для удаленного администрирования систем, важно понимать основы того, как он шифрует связь между **клиентом** и **сервером SSH**. Затем вы узнаете, как создать пару из открытого и закрытого ключей, чтобы соединения даже не подвергали риску пароли пользователей. Но сначала может быть полезно просмотреть некоторую основную информацию о командах и файлах конфигурации **SSH**.

Команды настройки SSH

Есть несколько **SSH-ориентированных** утилит, о которых вам нужно знать:

- **sshd** Сервис демонов; он должен быть запущен для получения входящих запросов клиента **Secure Shell**.
- **ssh-agent** Программа для хранения закрытых ключей, используемая для аутентификации алгоритма цифровой подписи **Digital Signature Algorithm** (**DSA**), эллиптической кривой **Elliptic Curve DSA** (**ECDSA**) и **Rivest, Shamir, Adleman (RSA).** Идея состоит в том, что команда **ssh-agent** запускается в начале **сеанса X** или сеанса входа в систему, а другие программы запускаются? как клиенты для программы **ssh-agent.**
- **ssh-add** Добавляет идентификаторы закрытого ключа агенту аутентификации **ssh-agent**.
- **ssh** Команда **Secure Shell, ssh**, является безопасным способом входа на удаленный компьютер, аналогично **Telnet** или **rlogin**. Основное использование этой команды обсуждалось в **Главе 2**. Чтобы это работало с аутентификацией на основе ключей, вам нужен закрытый ключ на клиенте и открытый ключ на сервере. Возьмите файл с открытым ключом, такой, как **id_rsa.pub** , созданный позже в этом разделе.

Скопируйте его на сервер. Поместите его в домашний каталог авторизованного пользователя в файле **~/.ssh/authorized_keys**.

- **ssh-keygen** Утилита, которая создает пары закрытых/открытых ключей для аутентификации **SSH**. Команда **ssh-keygen -t *keytype*** создаст пару ключей на основе протокола **DSA, ECDSA** или **RSA**.
- **ssh-copy-id** Сценарий, который копирует открытый ключ в целевую удаленную систему.

Файлы конфигурации клиента SSH

Системы, настроенные с использованием **SSH**, содержат файлы конфигурации в двух разных каталогах. Для локальной системы основные файлы конфигурации SSH хранятся в каталоге **/etc/ssh**. Но не менее важны файлы конфигурации в домашнем каталоге каждого пользователя в подкаталоге **~/.ssh/**.

Эти файлы настраивают, как данному пользователю разрешено подключаться к удаленным системам. Если включены ключи **DSA, ECDSA** и **RSA**, пользовательский каталог **~/.ssh/** содержит следующие файлы:

- **authorized_keys** Включает в себя список открытых ключей от удаленных пользователей. Пользователи с открытыми ключами шифрования в этом файле могут подключаться к удаленным системам. Пользователи системы и имена указаны в конце каждого открытого ключа, скопированного в этот файл.

- **id_dsa** Включает локальный закрытый ключ на основе алгоритма **DSA**.
- **id_dsa.pub** Включает локальный открытый ключ для пользователя на основе алгоритма DSA.
- **id_ecdsa** Включает локальный закрытый ключ на основе алгоритма **ECDSA**.
- **id_ecdsa.pub** Включает локальный открытый ключ для пользователя на основе алгоритм **ECDSA**.
- **id_rsa** Включает локальный закрытый ключ на основе алгоритма **RSA**.
- **id_rsa.pub** Включает локальный открытый ключ для пользователя на основе алгоритма **RSA**.
- **known_hosts** Содержит открытые ключи хоста от удаленных систем. При первом входе пользователя в систему ему предлагается принять открытый ключ удаленного сервера. На RHEL 7 протокол **ECDSA** по умолчанию используется для шифрования трафика. Соответствующий открытый ключ на удаленном сервере хранится в файле /etc/ssh/ssh_host_ecdsa_key.pub и добавляется клиентом в его локальный файл ~/.ssh/known_hosts.

Основы шифрования соединений

Базовое шифрование в компьютерных сетях обычно требует секретного ключа и открытого ключа. Принцип тот же, что и при обмене данными GPG, рассмотренном в главе 10. Владелец хранит закрытый ключ, а открытый ключ отправляется третьей стороне. Когда пара ключей настроена правильно, пользователь может зашифровать сообщение, используя свой закрытый ключ, в то время как третье лицо может расшифровать сообщение с помощью соответствующего открытого ключа. Это также работает в обратном порядке: третье лицо может зашифровать сообщение, используя открытый ключ получателя, в то время как получатель может расшифровать сообщение с помощью своего личного ключа. **Протокол SSH работает аналогичным образом**: сервер отправляет копию своего открытого ключа клиенту, и этот ключ используется клиентом для расшифровки трафика и установки безопасного канала связи.

Ключи шифрования основаны на случайных числах. Числа настолько велики (обычно 2048 бит для ключей RSA или более), что вероятность того, что кто-то проникнет в серверную систему, по крайней мере, с ПК, практически невозможна. Закрытые и открытые ключи шифрования основаны на согласованном наборе этих случайных чисел.

Частные ключи

Закрытый ключ должен быть защищён. Аутентификация на основе ключей основана на закрытом ключе, который доступен только владельцу этого ключа в подкаталоге **~/.ssh** домашнего каталога этого пользователя. Для аутентификации пользователя сервер отправляет клиенту «запрос», который представляет собой запрос на выполнение операции шифрования, которая требует знания закрытого ключа. Как только сервер получит ответ на свой запрос от клиента, он сможет расшифровать сообщение и доказать подлинность личности пользователя.

Публичные ключи

Публичный ключ - это общедоступный ключ. Открытые ключи предназначены для копирования в соответствующие подкаталоги **~/.ssh/** соответствующих пользователей в файле с именем **authorized_keys**.

В примере, показанном на **рисунке 4-7**, перечислены каталоги и файлы, связанные с использованием **SSH**.

РИСУНОК 4-7 Ключи в подкаталоге пользователя .ssh/

```
[michael@server1 ~]$ ls -l .ssh/
total 20
-rw-------. 1 michael michael 1822 Jan  7 21:43 authorized_keys
-rw-------. 1 michael michael  227 Sep 12 20:29 id_ecdsa
-rw-r--r--. 1 michael michael  186 Sep 12 20:29 id_ecdsa.pub
-rw-------. 1 michael michael 1679 Nov  7 18:24 id_rsa
-rw-r--r--. 1 michael michael  406 Nov  7 18:24 id_rsa.pub
-rw-r--r--. 1 michael michael  346 Jan  7 21:44 known_hosts
[michael@server1 ~]$
```

!!!!! EXAM Watch !!!!!
Большинство распространенных проблем с аутентификацией на основе ключей SSH связаны с правами доступа к файлам. Как показано на рисунке 4-7, разрешения для закрытых ключей установлены на 600, а для открытых ключей - на 644. Кроме того, разрешения для каталога ~/.ssh должны быть 700.
!!!!!!

Ключ похож на пароль, используемый для шифрования коммуникационных данных. Но это ни в коем случае не стандартный пароль. Представьте себе, что вы пытаетесь запомнить 1024-битное число, выраженное в шестнадцатеричном формате, показанном здесь:

3081 8902 8181 00D4 596E 01DE A012 3CAD 51B7
7835 05A4 DEFC C70B 4382 A733 5D62 A51B B9D6
29EA 860B EC2B 7AB8 2E96 3A4C 71A2 D087 11D0
E149 4DD5 1E20 8382 FA58 C7DA D9B0 3865 FF6E
88C7 B672 51F5 5094 3B35 D8AA BC68 BBEB BFE3
9063 AE75 8B57 09F9 DCF8 FFA4 E32C A17F 82E9
7A4C 0E10 E62D 8A97 0845 007B 169A 0676 E7CF
5713

Закрытый ключ похож, но вы должны держать его закрытым, иначе вся система выйдет из строя. Сохранение конфиденциальности означает, что никто не должен иметь доступа к серверным системам. Если ваш компьютер общедоступен, защитите свой закрытый ключ парольной фразой (паролем). Процедура установки ключевой фразы описана ниже. Не забудьте парольную фразу, иначе вам придется создать новую пару ключей и снова скопировать ваш открытый ключ во все целевые системы.

Настройка частной/публичной пары для аутентификации на основе ключей

Команда ssh-keygen используется для настройки пары **открытый/закрытый ключ**. Хотя он создает **ключ RSA** по умолчанию, его также можно использовать для создания ключа DSA или ECDSA. Например, некоторым пользователям могут понадобиться ключи DSA для соответствия определенным государственным стандартам США. Пример последовательности команд показан на **рисунке 4-8**.

РИСУНОК 4-8 Команда для генерации пары ключей SSH

```
[michael@server1 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/michael/.ssh/id_rsa):
Created directory '/home/michael/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/michael/.ssh/id_rsa.
Your public key has been saved in /home/michael/.ssh/id_rsa.pub.
The key fingerprint is:
3f:63:1e:4e:0e:82:f1:e9:2c:c3:2b:b8:d7:7e:57:06 michael@server1.example.net
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|        E        |
|     .   S.      |
|      + . .o     |
| . o. + .oB      |
|. o =o...B +     |
|.o oo=o.  +      |
+-----------------+
[michael@server1 ~]$
```

Как показано на рисунке, команда запрашивает дополнительную фразу-пароль для защиты закрытого ключа. После подтверждения идентичной парольной фразы закрытый ключ сохраняется в файл **id_rsa**, а соответствующий открытый ключ хранится в файле **id_rsa.pub**. Оба файла для пользователя **michael** хранятся в каталоге **/home/michael/.ssh**.

При желании вы можете установить ключи RSA с большим количеством битов. В нашем тестировании мы смогли довольно быстро настроить пары ключей длиной до 8192 бит даже в системе виртуальных машин с одним виртуальным процессором.

Команда, которая запускает процесс

$ ssh-keygen -b 8192

В качестве альтернативы, если нужен ключ DSA, может помочь следующая команда. Разрешены только 1024-битные ключи DSA. Процесс после этой команды такой же, как показано на **рисунке 4-8.**

$ ssh-keygen -t dsa

Следующим шагом является передача открытого ключа в удаленную систему. Это может быть один из серверов, которыми вы управляете. Если вы хотите передать этот открытый ключ по сети (один раз на соединение) может работать следующая команда:

$ ssh-copy-id -i .ssh/id_rsa.pub michael@tester1.example.com

Строго говоря, команда **ssh-copy-id без опции -i** по умолчанию передает последний созданный открытый ключ. Предыдущая команда автоматически добавляет указанный локальный ключ RSA в конец удаленного файла **~/.ssh/authorized_keys.** В этом случае этот файл можно найти в каталоге **/home/michael**. Конечно, вы можете заменить IP-адрес именем хоста.

!!!!! EXAME Watch !!!!!
Иногда, после копирования пары ключей в удаленную систему, вы можете получить сообщение об ошибке «агент признал невозможность подписи с помощью ключа», после чего появится запрос пароля при попытке войти в систему. Чтобы устранить эту проблему, выйдите из консоли или GUI и войдите в систему. В большинстве случаев команда ssh запросит парольную фразу.
!!!!!

После этого вы сможете сразу подключиться к этой удаленной системе. В предыдущем случае подходящая команда является одной из следующих:

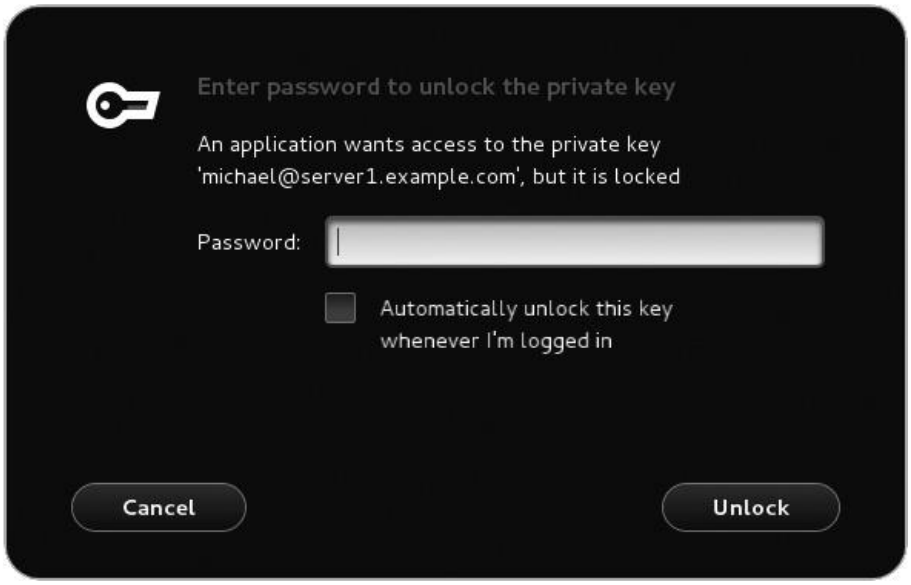
$ ssh -l michael tester1.example.com
$ ssh michael@tester1.example.com

При запуске на консоли команда **ssh** использует следующую подсказку для ключевой фразы:

Enter passphrase for key '/home/michael/.ssh/id_rsa'

При запуске в командной строке на основе графического интерфейса появляется окно, аналогичное показанному на **рис. 4-9**.

РИСУНОК 4-9 запрос пароля



ЦЕЛЬ СЕРТИФИКАЦИИ 4.05

Учебник по Linux с улучшенной безопасностью

Security-Enhanced Linux (**SELinux**) был разработан Агентством национальной безопасности США для обеспечения уровня обязательного контроля доступа для Linux. Это выходит за рамки дискреционного контроля доступа, связанного с правами доступа к файлам и списками ACL. По сути, **SELinux** обеспечивает соблюдение правил безопасности в ядре операционной системы. Это ограничивает ущерб в случае нарушения безопасности. Например, если системная учетная запись, связанная со службой FTP, скомпрометирована, **SELinux** усложняет использование этой учетной записи для компрометации других служб.

Основные характеристики SELinux

Модель безопасности **SELinux** основана на субъектах, объектах и действиях (**subjects, objects, и actions**). Субъект (**subjects**) - это процесс, такой как работающая команда или приложение, такое как работающий веб-сервер Apache. Объект (**objects**) - это файл, устройство, сокет или вообще любой ресурс, к которому субъект может получить доступ. Действие (**actions**) - это то, что может быть сделано субъектом с объектом.

SELinux назначает различные **контексты** объектам. Контекст (**context**) - это просто метка, которая используется политикой безопасности **SELinux**, чтобы определить, разрешено или нет действие субъекта над объектом.

Например, процесс веб-сервера Apache может принимать такие объекты, как файлы веб-страниц, и отображать их для просмотра клиентами всего мира. Это действие обычно разрешено в реализации **SELinux на RHEL 7**, если объектные файлы имеют соответствующий контекст **SELinux**.

Контексты, связанные с SELinux, детализированы. Другими словами, если хакер «черной шляпы» проникнет и захватит ваш веб-сервер, контексты SELinux не позволят взломщику использовать это нарушение для проникновения в другие сервисы.

Чтобы увидеть контекст определенного файла, выполните команду **ls -Z**. В качестве примера рассмотрим, что эта команда делает **на рис. 4-10,** так как она отображает контексты безопасности в одной из директорий **/root** этой книги автора.

Как отмечалось в начале этой главы, пять задач касаются SELinux на экзамене RHCSA. Вы узнаете, как достичь этих целей в следующих разделах.

SELinux Status

Как указано в целях RHCSA, вам необходимо знать, как «установить принудительные и разрешающие режимы для SELinux»(set **enforcing** and **permissive** modes for SELinux). Для SELinux существует три доступных режима: принудительный (**enforcing**), разрешительный (**permissive**) и отключенный (**disabled**). Принудительный и отключенный режимы говорят сами за себя.

SELinux в разрешающем (**permissive**) режиме означает, что все нарушенные правила SELinux регистрируются, но нарушение не останавливает никаких действий.

Если вы хотите изменить режим SELinux по умолчанию, измените директиву **SELINUX** в файле **/etc/selinux/config**, как показано в **Таблице 4-10**. При следующей перезагрузке изменения будут применены к системе.

!!!! On the Job !!!!
В RHEL 6 переменная конфигурации SELINUX была определена в файле /etc/sysconfig/selinux. В RHEL 7 /etc/sysconfig/selinux является символической ссылкой, указывающей на файл конфигурации /etc/selinux/config.
!!!!!

РИСУНОК 4-10. Контекст безопасности SELinux для разных файлов

```
[root@server1 ~]# ls -Z
-rw-------. root root system_u:object_r:admin_home_t:s0 anaconda-ks.cfg
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 backup
-rwxr--r--. root root unconfined_u:object_r:admin_home_t:s0 Ch3Lab2
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 Ch3Lab2testfile
-rwxr--r--. root root unconfined_u:object_r:admin_home_t:s0 Ch3Lab3
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 Ch3Lab3testfile
-rwxr--r--. root root unconfined_u:object_r:admin_home_t:s0 Ch3Lab4
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 Ch3Lab4testfile
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 Desktop
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 Documents
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 Downloads
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 hosts
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 ifcfg-eth0
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 ifcfg-System_eth0
-rw-r--r--. root root system_u:object_r:admin_home_t:s0 install.log
-rw-r--r--. root root system_u:object_r:admin_home_t:s0 install.log.syslog
-rw-------. root root unconfined_u:object_r:admin_home_t:s0 ks.cfg
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 Music
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 Pictures
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 Public
-rw-r--r--. root root system_u:object_r:net_conf_t:s0  route-System_eth0
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 Templates
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 Videos
[root@server1 ~]#
```

Если **SELinux** настроен в принудительном режиме, он защищает системы одним из двух способов: в целевом режиме или в режиме **mls**. По умолчанию используется целевая политика, которая позволяет вам детально настроить то, что защищено **SELinux**. Напротив, **MLS** идет дальше, используя модель **Bell-La Padula**, разработанную для Министерства обороны США. Эта модель, как предлагается в файле **/etc/selinux/targeted/setrans.conf,** поддерживает уровни безопасности между уровнями c0 и c3. Хотя уровень c3 указан как «Совершенно секретно», диапазон доступных уровней доходит до c1023. Такие мелкозернистые уровни секретности еще полностью не разработаны. **Если вы хотите изучить MLS, установите RPM selinux-policy-mls**.

ТАБЛИЦА 4-10 Стандартные директивы конфигурации в /etc/selinux/config

Директива	Описание
SELINUX	Базовый статус **SELinux**; может быть установлено принудительное, разрешающее или отключенное.
SELINUXTYPE	Определяет уровень защиты; по умолчанию установлен на целевой, где защита ограничена выбранными «целевыми» услугами. Альтернативой является mls, которая связана с многоуровневой безопасностью (MLS).

!!!!! On the Job !!!!
Если вы просто хотите поэкспериментировать с SELinux, настройте его в разрешающем режиме. Он будет регистрировать любые нарушения, не останавливая ничего. Его легко настроить с помощью инструмента администрирования SELinux, или вы можете установить SELINUX=permissive в /etc/selinux/config.. Если служба auditd работает, нарушения регистрируются в файле audit.log в /var/log/audit Audit Directory. Просто помните, что Red Hat, вероятно, хочет, чтобы кандидаты настраивали SELinux в принудительном режиме во время экзаменов.
!!!!!

Конфигурация SELinux в командной строке

Хотя SELinux все еще находится в активной разработке, он стал намного более полезным с выпусками RHEL 6 и RHEL 7. Тем не менее, учитывая сложность, связанную с SELinux, он может быть более эффективным для системных инженеров, которые не очень знакомы с ним, чтобы использовать инструмент администрирования SELinux для настройки параметров SELinux.

В следующих разделах показано, как вы можете настраивать и управлять SELinux из интерфейса командной строки. Однако, поскольку проще продемонстрировать все возможности SELinux с помощью инструментов с графическим интерфейсом, подробное обсуждение таких возможностей будет дано далее в этой главе.

Настройте основные параметры SELinux

Есть несколько важных команд, которые можно использовать для просмотра и настройки основных параметров **SELinux**. Чтобы увидеть текущее состояние SELinux, **запустите команду getenforce**; он возвращает один из трех очевидных параметров: принудительный, разрешающий или отключенный (**enforcing, permissive, or disabled**). Команда **sestatus** предоставляет больше информации, с выводом, подобным следующему.

SELinux status: **enabled**
SELinuxfs mount: **/sys/fs/selinux**
SELinux root directory: **/etc/selinux**
Loaded policy name: **targeted**
Current mode: **enforcing**
Mode from config file: **enforcing**
Policy MLS status: **enabled**
Policy deny_unknown status: **allowed**
Max kernel policy version: **28**

Вы можете изменить текущий статус **SELinux** с помощью команды **setenforce**; варианты просты:

 # setenforce enforcing
 # setenforce permissive

Это изменяет логическое значение **/sys/fs/selinux/enforce**. Для логических значений вы **можете заменить 1 и 0** соответственно на **enforcing** и **permissive.** Чтобы сделать это изменение постоянным, вам нужно изменить переменную SELINUX в файле **/etc/selinux/config**. Однако для внесения изменений в подробные логические выражения SELinux требуются другие команды.

В качестве альтернативы, если SELinux по какой-то причине отключен, вывод будет

SELinux status: **disabled**

В этом случае команда **setenforce** не будет работать. Вместо этого вам нужно установить **SELINUX=enforcing** в файле **/etc/selinux/config**. И это требует перезагрузки системы для «перемаркировки» всех файлов, когда метки SELinux применяются к каждому файлу в локальной системе.

!!!!!! On the Job !!!!!
Если SELinux отключен, перезагрузка системы может занять несколько минут после установки SELinux в принудительном режиме. Однако этот процесс занимает меньше времени, чем в предыдущих выпусках RHEL.
!!!!!!!

Настройте обычных пользователей для SELinux

Чтобы просмотреть статус текущих пользователей SELinux, выполните команду **semanage login -l**. На основании установки по умолчанию RHEL 7 это приводит к следующему выводу:

Login Name	SELinux User	MLS/MCS Range	Service
__default__	**unconfined_u**	**s0-s0:c0.c1023**	*****
root	**unconfined_u**	**s0-s0:c0.c1023**	*****
system_u	**system_u**	**s0-s0:c0.c1023**	*****

Другими словами, обычные пользователи по умолчанию имеют тот же пользовательский контекст **SELinux**, что и пользователь **root**. Для подтверждения запустите **команду id -Z** от имени обычного пользователя. Без изменений это приводит к следующему выводу, который предполагает, что пользователь не ограничен никакими настройками **SELinux:**

unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

Предыдущая строка определяет то, что называется меткой в жаргоне SELinux. Метка состоит из нескольких строк контекста, разделенных столбцом: пользовательский контекст (который заканчивается на **_u**), контекст роли (который заканчивается на **_r**), контекст типа (который заканчивается на **_t**), контекст чувствительности, и набор категорий. Правила целевой политики, которая является политикой SELinux по умолчанию в RHEL 7, в основном связаны с контекстом типа **(_t)**.

Хотя это может и не быть требованием к экзамену, обычные пользователи должны ограничиваться SELinux. Когда учетные записи пользователей скомпрометированы, и они будут скомпрометированы, вы хотите, чтобы любой ущерб, который может быть причинен, ограничен правилами SELinux. В следующем примере указывается правило ограничения, которое добавляет **(-a)** обычного пользователя **michael**, указывая **(-s) контекст user_u** для ограничения:

 # semanage login -a -s user_u michael

Роль **user_u** не должна иметь возможности запускать команды **su** и **sudo**, описанные в главе 8. При желании вы можете изменить процесс с помощью команды **semanage -d michael**. Поскольку роли пользователей все еще находятся в стадии разработки, вам следует сосредоточиться на доступных пользовательских контекстах, перечисленных в последней документации Red Hat, как показано в **Таблице 4-11**.

Еще одним часто встречающимся «**пользовательским (user)**» контекстом является **system_u**, который обычно не применяется к обычным пользователям. Это обычный пользователь, который можно увидеть в выходных данных команды **ls -Z** для системных файлов и файлов конфигурации.

Когда пользовательская роль изменяется, она не вступает в силу до следующего входа в систему. Например, если мы изменим роль для пользователя **michael** на **user_u** в командной строке на основе графического интерфейса, это изменение не вступит в силу, пока мы не выйдем из системы и не войдем обратно в GUI. Если вы попробуете это в своей системе, вы больше не сможете запускать какие-либо инструменты административного конфигурирования, и у вас не будет доступа к командам **sudo** и **su**.

В некоторых сетях вы можете изменить роль будущих пользователей на **user_u**. Если вы не хотите, чтобы обычные пользователи работали с инструментами администрирования, вы можете внести это изменение для будущих пользователей по умолчанию с помощью следующей команды:

 # semanage login -m -S targeted -s "user_u" -r s0 __default__

ТАБЛИЦА 4-11 Опции для ролей пользователя SELinux

Пользовательский контекст	Особенности
guest_u	Нет графического интерфейса, нет сети, нет доступа к командам su или sudo, нет выполнения файлов в **/home** или **/tmp**
xguest_u	Графический интерфейс, работа в сети только через веб-браузер Firefox, без выполнения файлов в **/home** или **/tmp**
user_u	GUI и сеть доступны
staff_u	Графический интерфейс, сеть и команда **sudo** доступны
sysadm_u	Доступны графический интерфейс, работа в сети и команды **sudo** и **su**
unconfined_u	Полный доступ к системе

Эта команда изменяет **(-m)** целевое хранилище политик **(-S),** используя **SELinux user (-s) user_u**, с диапазоном **MLS s0 (-r)** для пользователя по умолчанию. Здесь «**__default__**» включает в себя два символа подчеркивания с каждой стороны слова. Пока **user_u** действует для пользователя **SELinux** по умолчанию, обычные пользователи не будут иметь доступа к использованию инструментов администрирования или команд, таких как **su** и **sudo**. Следующая команда полностью изменяет процесс:

 # semanage login -m -S targeted -s "unconfined_u" -r s0-s0:c0.c1023 __default__

Требуется полный диапазон **MLS (s0-s0: c0.c1023)**, поскольку пользователь не **unconfined_u** обычно не ограничен ограничениями **MLS**.

!!!! EXAM Watch !!!!!
Политика MLS добавляет сложности SELinux. Целевая политика по умолчанию с соответствующими логическими значениями и контекстами файлов обычно обеспечивает более чем достаточную безопасность.
!!!!!!

Управление логическими настройками SELinux

Большинство настроек **SELinux** являются логическими - другими словами, они активируются и деактивируются, **устанавливая их в 1 или 0 соответственно**. После установки логические значения могут быть получены из каталога **/sys/fs/selinux/booleans**. Одним простым примером является **selinuxuser_ping**, который обычно **равен 1**, что позволяет пользователям запускать команды **ping** и **traceroute**. Многие из этих настроек **SELinux** связаны с конкретными службами **RHCE** и будут рассмотрены во второй половине этой книги.

Эти настройки можно прочитать с помощью **getsebool** и изменить с помощью команд **setsebool**. Например, следующий вывод команды **getsebool user_exec_content** подтверждает, что **SELinux** позволяет пользователям выполнять сценарии либо в своих домашних каталогах, либо в каталоге **/tmp**:

user_exec_content --> on

Это значение по умолчанию применяется к пользователям **SELinux user_u**. Другими словами, с этим логическим значением такие пользователи могут создавать и выполнять сценарии в отмеченных каталогах. Это логическое значение может быть отключено либо временно, либо таким образом, чтобы пережить перезагрузку. Один из способов сделать это с помощью команды **setsebool**. Например, следующая команда отключает указанное логическое значение до перезагрузки системы:

 # setsebool user_exec_content off

Вы можете выбрать замену **= 0** для **off** в команде. Поскольку это логическая настройка, эффект тот же: флаг отключен. Однако **ключ -P** необходим для того, чтобы изменение логического параметра пережило перезагрузку системы. Имейте в виду, что изменения не вступят в силу до следующего раза, когда указанный пользователь фактически войдет в связанную систему.

Полный список доступных логических значений доступен в выводе команды **getsebool -a.**

Для получения дополнительной информации о каждом логическом значении выполните команду **semanage boolean -l**. Хотя выходные данные включают описания всех доступных логических значений, это база данных, в которой можно искать с помощью команды **grep**.

Перечислите и определите контексты файлов SELinux

Если вы включили **SELinux**, команда **ls -Z** перечисляет текущие контексты файла **SELinux**, как показано ранее на **рисунке 4-10**. В качестве примера возьмем соответствующий вывод для файла **anaconda-ks.cfg** из каталога **/root:**

-rw-------. root root system_u:object_r:admin_home_t:s0 anaconda-ks.cfg

Вывод включает в себя данные о владельце и праве **ugo/rwx**. Он также определяет четыре элемента безопасности SELinux: **пользователь, роль, тип и уровень MLS (user, role, type, and MLS level**) для отмеченного файла. Как правило, пользователь SELinux, связанный с файлом, **имеет имя system_u** или **undefined_u**, и это, как правило, не влияет на доступ. В большинстве случаев файлы связаны с **object_r,** ролью объекта для файла. Конечно, возможно, что будущие версии целевой политики SELinux будут включать более детальные параметры для пользователя и роли.

Ключевым контекстом файла **является тип**, в данном случае **admin_home_t**. Когда вы настроили серверы FTP и HTTP в главе 1, вы изменили тип настроенного каталога и файлов в нем, чтобы он соответствовал типу общих файлов по умолчанию из этих служб с помощью **команды chcon**.

Например, чтобы настроить нестандартный каталог для FTP-сервера, убедитесь, что контекст соответствует стандартному каталогу FTP. Рассмотрим следующую команду:

 # ls -Z /var/ftp/
drwxr-xr-x. root root system_u:object_r:public_content_t pub

Контекстами являются системный пользователь (**system_u**) и системные объекты (**object_r**) для общего доступа к типу с public (**public_content_t**). Если вы создадите другой каталог для службы FTP, вам нужно будет назначить тот же контекст безопасности для этого каталога. Например, если вы создаете каталог **/ftp** от имени пользователя root и запускаете команду **ls -Zd /ftp**, вы увидите контексты, связанные с каталогом **/ftp**, как показано ниже:

drwxr-xr-x. root root unconfined_u:object_r:root_t /ftp

Чтобы изменить контекст, используйте команду **chcon**. Если есть подкаталоги, вы должны убедиться, что изменения внесены **рекурсивно с ключом -R**. В этом случае, чтобы изменить пользователя и ввести контексты для соответствия **/var/ftp**, выполните следующую команду:

 # chcon -R -u system_u -t public_content_t /ftp

Если вы хотите поддерживать загрузку на свой FTP-сервер, вам придется назначить контекст другого типа, в частности **public_content_rw_t**. Это соответствует следующей команде:

 # chcon -R -u system_u -t public_content_rw_t /ftp

В главе 1 вы использовали другой вариант команды **chcon**. Чтобы использовать этот урок, следующая команда использует пользователя, роль и контекст из каталога **/var/ftp** и применяет изменения рекурсивно:

 # chcon -R --reference /var/ftp /ftp

Но подождите, что произойдет, если файловая система будет перемаркирована? Изменения, сделанные с помощью **chcon**, не выдержат перемаркировки файловой системы, потому что все контексты файла будут сброшены к значениям по умолчанию, определенным в политике SELinux. Следовательно, нам нужен способ изменить правила, которые определяют контекст файла по умолчанию для каждого файла. Эта тема будет рассмотрена в следующем разделе.

!!!!! On the Job !!!!
Использование *restorecon* является предпочтительным способом контекстов изменения файлов, поскольку он устанавливает контексты к значениям, настроенным в политике SELinux. Команда *chcon* может изменить контексты файла для любого значения, переданного в качестве аргумента, но это изменение может не выдержать перемаркировку файловой системы, если контекст отличается от значения по умолчанию, определенного в политике SELinux. Поэтому, чтобы избежать ошибок, вы должны изменить контексты в политике SELinux с *semanage fcontext* и использования *restorecon* в контексте изменения файлов.
!!!!!

Восстановить контексты файлов SELinux

Контексты по умолчанию настраиваются в **/etc/selinux/target/contexts/files/file_contexts**. Если вы допустили ошибку и хотите восстановить исходные параметры SELinux для файла, команда **restorecon** восстанавливает эти параметры на основе файла конфигурации **file_contexts**. Однако значения по умолчанию в каталоге могут отличаться. Например, следующая команда (с **ключом -F**, принудительно изменяющим все контексты, а не только контекст типа) приводит к другому набору контекстов для каталога **/ftp:**

 # restorecon -F /ftp
 # ls -Zd /ftp
drwxr-xr-x. root root system_u:object_r:default_t ftp

РИСУНОК 4-11 Определения контекста SELinux

```
/var/ftp(/.*)?                                  all files          system_u:object_r:public_content_t:s0
/var/ftp/bin(/.*)?                              all files          system_u:object_r:bin_t:s0
/var/ftp/etc(/.*)?                              all files          system_u:object_r:etc_t:s0
/var/ftp/lib(/.*)?                              all files          system_u:object_r:lib_t:s0
/var/ftp/lib/ld[^/]*\.so(\.[^/]*)*              regular file       system_u:object_r:ld_so_t:s0
/var/games(/.*)?                                all files          system_u:object_r:games_data_t:s0
/var/imap(/.*)?                                 all files          system_u:object_r:cyrus_var_lib_t:s0
/var/kerberos/krb5kdc(/.*)?                     all files          system_u:object_r:krb5kdc_conf_t:s0
/var/kerberos/krb5kdc/from_master.*             all files          system_u:object_r:krb5kdc_lock_t:s0
/var/kerberos/krb5kdc/kadm5\.keytab             regular file       system_u:object_r:krb5_keytab_t:s0
/var/kerberos/krb5kdc/principal.*               all files          system_u:object_r:krb5kdc_principal_t:s0
/var/kerberos/krb5kdc/principal.*\.ok           all files          system_u:object_r:krb5kdc_lock_t:s0
```

Вы можете заметить, что пользовательский контекст отличается от того, когда был создан каталог **/ftp**. Это связано с первой строкой в вышеупомянутом файле **file_contexts**, который применяет отмеченные контексты:

/.* system_u: object_r: default_t: s0

Вы также можете перечислить все правила контекстов файлов по умолчанию в **file_contexts** с помощью команды **semanage fcontext -l**. См. **Рисунок 4-11** с выдержкой из выходных данных.

Как видите, определения контекста SELinux используют регулярные выражения, такие как:

(/.*)?

Предыдущее регулярное выражение соответствует **символу** /, за которым следует произвольное количество символов **(.*).** Этот символ (**?)** означает, что все регулярное выражение в скобках может совпадать с нулем или один раз. Следовательно, общий результат - совпадение с /, за которым следует произвольное количество символов или ничего. Это регулярное выражение широко используется для сопоставления каталога и всех файлов в нем.

Например, регулярное выражение, соответствующее каталогу **/ftp** и всем файлам в нем, задается следующим образом:

/ftp(/.*)?

Используя это регулярное выражение, мы можем определить правило политики SELinux, которое присваивается

Каталог **/ftp** и все файлы в нем - контекст по умолчанию. Это можно сделать с помощью команды **semanage fcontext -a**. Например, следующая команда назначает контекст типа по умолчанию **public_content_t** каталогу **/ftp** и всем файлам в нем:

 # semanage fcontext -a -t public_content_t '/ftp(/.*)?'

После того как вы определили новый контекст политики по умолчанию для пути к файловой системе, вы можете запустить команду **restorecon**, чтобы установить для контекстов соответствующие значения политики по умолчанию.

Следующая команда восстанавливает контекст **рекурсивно (-R)** до значения **public_content_t,** определенного ранее:

 # restorecon -RF /ftp
 # ls -Zd /ftp
drwxr-xr-x. root root system_u:object_r:public_content_t ftp

Определите контексты процесса SELinux

Как обсуждалось в главе 9, команда **ps** выводит список запущенных в данный момент процессов. В системе SELinux существуют контексты для каждого запущенного процесса. Чтобы увидеть эти контексты для всех процессов, работающих в настоящее время, выполните команду **ps -eZ**, которая перечисляет каждый процесс (**every**) **(-e)** SELinux контекст каждого **(-Z)** процесса. **Рисунок 4-12** включает в себя различные выдержки этой команды с нашей системы.

Хотя пользователь и роль меняются не часто, тип процесса широко варьируется, часто в зависимости от цели выполняемого процесса. Например, в нижней части рисунка видно, как демон **Avahi (avahi-daemon)** соответствует типу **avini_t** SELinux. Вы должны быть в состоянии определить, как, по крайней мере, некоторые другие типы SELinux соответствуют связанной службе.

Другими словами, хотя существует большое разнообразие типов SELinux, они согласуются с запущенным процессом.

РИСУНОК 4-12 SELinux контексты безопасности различных процессов

```
system_u:system_r:kernel_t:s0     486 ?        00:00:00 rpciod
system_u:system_r:syslogd_t:s0    499 ?        00:00:00 systemd-journal
system_u:system_r:lvm_t:s0        502 ?        00:00:00 lvmetad
system_u:system_r:udev_t:s0-s0:c0.c1023 517 ?  00:00:00 systemd-udevd
system_u:system_r:kernel_t:s0     537 ?        00:00:00 vballoon
system_u:system_r:kernel_t:s0     562 ?        00:00:00 kvm-irqfd-clean
system_u:system_r:kernel_t:s0     569 ?        00:00:00 hd-audio0
system_u:system_r:kernel_t:s0     588 ?        00:00:00 xfs-data/vda1
system_u:system_r:kernel_t:s0     591 ?        00:00:00 xfs-conv/vda1
system_u:system_r:kernel_t:s0     592 ?        00:00:00 xfs-cil/vda1
system_u:system_r:kernel_t:s0     594 ?        00:00:00 xfsaild/vda1
system_u:system_r:auditd_t:s0     600 ?        00:00:00 auditd
system_u:system_r:audisp_t:s0     608 ?        00:00:00 audispd
system_u:system_r:audisp_t:s0     613 ?        00:00:00 sedispatch
system_u:system_r:alsa_t:s0       627 ?        00:00:00 alsactl
system_u:system_r:firewalld_t:s0  629 ?        00:00:00 firewalld
system_u:system_r:avahi_t:s0      632 ?        00:00:00 avahi-daemon
system_u:system_r:syslogd_t:s0    633 ?        00:00:00 rsyslogd
system_u:system_r:tuned_t:s0      634 ?        00:00:00 tuned
system_u:system_r:abrt_t:s0-s0:c0.c1023 636 ?  00:00:00 abrtd
system_u:system_r:abrt_watch_log_t:s0 637 ?    00:00:00 abrt-watch-log
system_u:system_r:abrt_watch_log_t:s0 640 ?    00:00:00 abrt-watch-log
system_u:system_r:avahi_t:s0      650 ?        00:00:00 avahi-daemon
:
```

Диагностика и устранение нарушений политики SELinux

Если есть проблема, SELinux работает в принудительном режиме, и вы уверены, что нет проблем с целевой службой или приложением, не отключайте SELinux! Red Hat облегчает управление и устранение неисправностей. Согласно Red Hat, двумя основными причинами проблем, связанных с SELinux, являются контексты и логические настройки.

SELinux Проверка, аудит.

Проблемы с SELinux должны быть задокументированы в соответствующем файле журнала **audit.log**, в **/var/log/Audit Directory**. Файл может сбивать с толку, особенно при первом чтении. Доступен ряд инструментов, помогающих расшифровать этот журнал.

Во-первых, команда аудита поиска (**ausearch**) может помочь отфильтровать определенные типы проблем. Например, следующая команда перечисляет все события SELinux, связанные **с использованием команды sudo**:

 # ausearch -m avc -c sudo

Такие события известны как сообщения **Access Vector Cache (-m avc)**; **-c** позволяет указать имя, обычно используемое в журнале, например **httpd** или **su**. Если вы

экспериментировали с пользователем SELinux **user_u**, описанным ранее в этой главе, в файле **AuditLog** должно быть несколько связанных сообщений.

Даже для большинства администраторов вывод по-прежнему многословен. Тем не менее, он должен включать идентификационную информацию, такую как проверенный идентификатор пользователя (показанный как auid), который может помочь вам идентифицировать пользователя-нарушителя. Возможно, пользователю нужен такой доступ, или, возможно, его учетная запись была взломана. В любом случае предупреждение может заставить вас уделять больше внимания этой учетной записи.

Напротив, команда **sealert -a /var/log/audit/audit.log** может обеспечить большую ясность. Выдержка показана на **рисунке 4-13**.

РИСУНОК 4-13 Оповещение SELinux

```
SELinux is preventing /usr/bin/su from using the setuid capability.

*****  Plugin catchall_boolean (89.3 confidence) suggests   ******************

If you want to allow user  to use ssh chroot environment.
Then you must tell SELinux about this by enabling the 'selinuxuser_use_ssh_chroot'
 boolean.
You can read 'user_selinux' man page for more details.
Do
setsebool -P selinuxuser_use_ssh_chroot 1

*****  Plugin catchall (11.6 confidence) suggests   **************************

If you believe that su should have the setuid capability by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# grep su /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp


Additional Information:
Source Context                user_u:user_r:user_t:s0
Target Context                user_u:user_r:user_t:s0
Target Objects                 [ capability ]
Source                        su
Source Path                   /usr/bin/su
Port                          <Unknown>
Host                          <Unknown>
Source RPM Packages           sudo-1.8.6p7-11.el7.x86_64
Target RPM Packages
Policy RPM                    selinux-policy-3.12.1-153.el7_0.13.noarch
Selinux Enabled               True
Policy Type                   targeted
Enforcing Mode                Enforcing
:
```

Проблемы с маркерами и контекстом SELinux

Рассматривая **рисунок 4-13** и концепции SELinux, описанные выше, вы можете задаться вопросом, если пользователю разрешено запускать команду **su**. Если бы проблема была в файле **/etc/sudoers**, описанный в **главе 8**, может даже не появиться предупреждающее сообщение SELinux. Поэтому вам следует обратить внимание на исходный и целевой контексты. Поскольку они совпадают, контекст файла не является проблемой.

В процессе исключения это указывает на пользовательский контекст, описанный ранее, как проблема. UID соответствующего пользователя должен быть указан позже в файле в разделе «Необработанные сообщения аудита»(Raw Audit Messages). Если пользователю требуется доступ к командам **su** и **sudo**, вам следует изменить роль этого пользователя с помощью описанной ранее командой **semanage login**. В противном случае пользователь может просто экспериментировать с Linux. Любой доступ к команде **sudo** будет задокументирован в файле **/var/log/secure log**.

Булевы проблемы SELinux

После деактивации логического значения **user_exec_content**, описанного ранее, мы создали простой скрипт с именем **script1** для пользователя, управляемого меткой **user_u**. Сделав этот скрипт исполняемым, мы попытались запустить его с помощью команды **/home/examprep/script1**. Несмотря на то, что у этого пользователя был файл с установленными правами доступа, эта попытка привела к следующему сообщению:

-bash: /home/examprep/script1: Permission denied

Это привело к выдержке из журнала, показанной на **рисунке 4-14**. Обратите внимание на раздел в верхней части; в нем явно указана команда, необходимая для решения проблемы. Как администратор, вы должны решить, должны ли такие пользователи иметь возможность выполнять свои собственные сценарии. Если это так, то указанная команда решит проблему.

РИСУНОК 4-14 Предупреждение SELinux и решение

```
------------------------------------------------------------------------------

SELinux is preventing /usr/bin/bash from execute access on the file .

*****  Plugin catchall_boolean (89.3 confidence) suggests   ******************

If you want to allow user to exec content
Then you must tell SELinux about this by enabling the 'user_exec_content' boolean.
You can read 'user_selinux' man page for more details.
Do
setsebool -P user_exec_content 1

*****  Plugin catchall (11.6 confidence) suggests   **************************

If you believe that bash should be allowed execute access on the  file by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# grep bash /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp


Additional Information:
Source Context                user_u:user_r:user_t:s0
Target Context                unconfined_u:object_r:user_home_t:s0
Target Objects                 [ file ]
Source                        bash
:
```

Инструмент администрирования GUI SELinux

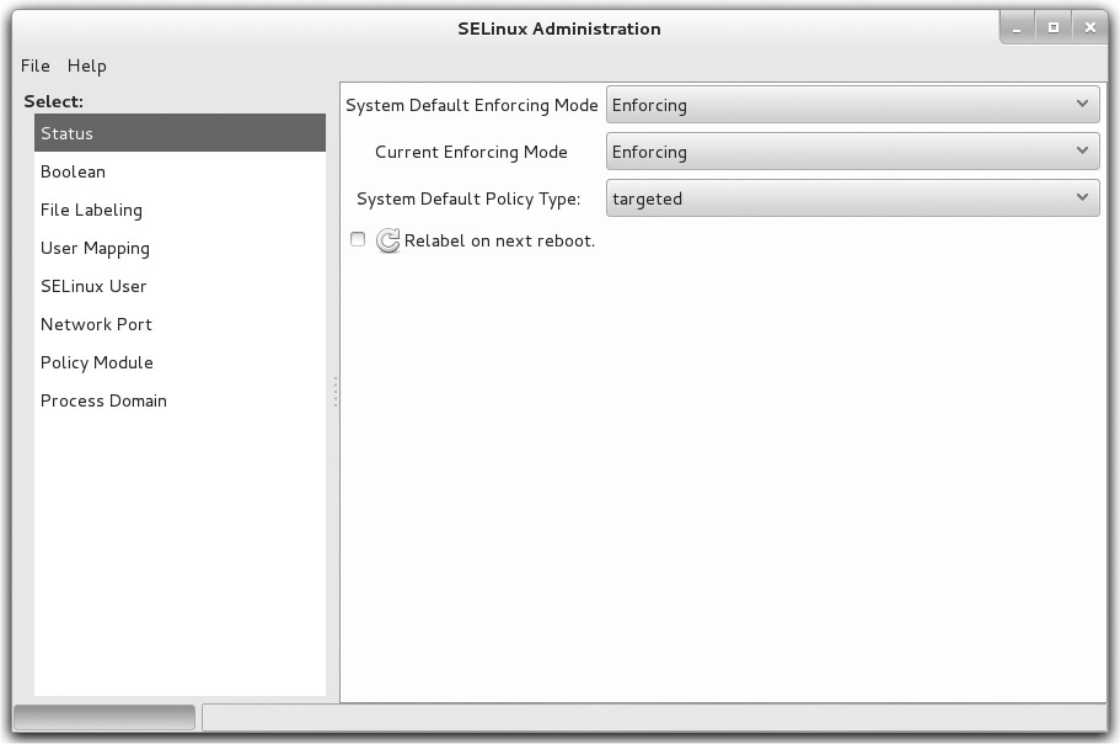
Если вы потратили время на изучение SELinux из командной строки, этот раздел должен стать просто обзором. Для многих пользователей самый простой способ изменить настройки SELinux - это инструмент администрирования SELinux, который можно запустить с помощью команды **system-config-selinux**. Как показано на **рисунке 4-15**, он начинается с базового представления о состоянии SELinux в локальной системе, отражая некоторую информацию, отображаемую в выходных данных команды **sestatus**.
Как вы можете видеть, есть опции, помеченные по умолчанию **Enforcing Mode** и **Current Enforcing Mode**, которые вы можете установить в **Enforcing**, **Permissive** или **Disabled**. Хотя в центре внимания SELinux находится целевая политика, MLS также доступен, если вы установите пакет **selinux-policy-mls**. Как правило, вам не нужно активировать опцию **Relabel On Next Reboot**, если вы не изменили тип политики по умолчанию.

На левой панели окна средства управления SELinux, описанного в следующих разделах, есть несколько категорий. В половине этой книги RHCE вы вернетесь к этому инструменту, уделив больше внимания логическим настройкам.

Логические настройки SELinux

В инструменте администрирования SELinux щелкните **Boolean** на левой панели. Прокрутите доступные модули. Как вы можете видеть, политика SELinux может быть изменена различными способами категории, некоторые из которых относятся к административным функциям, другие к конкретным услугам. Выбранное количество этих опций показано на **рисунке 4-16**. Любые сделанные вами изменения отражаются в логических переменных в каталоге **/sys/fs/selinux/booleans**. Категории модуля, представляющие интерес для экзамена RHCSA, включают в себя **cron**, **mount**, **virt** и эту общую категорию: **unknown**. Список выбранных логических значений включен в **таблицу 4-12.** Логические значения отображаются в порядке, указанном в инструменте управления SELinux.

РИСУНОК 4-15 Статус SELinux в инструменте администрирования



Маркировка файлов

Вы можете изменить метки по умолчанию, связанные с файлами, некоторые из которых описаны ранее в этой главе (и в других главах, обсуждающих контексты SELinux). Некоторые параметры показаны на **рисунке 4-17**. Любые изменения на этом экране записываются в файл **file_contexts.local** в каталоге **/etc/selinux/target/contexts/files**.

Отображение пользователей

Раздел «Сопоставление пользователей»(**User Mapping**) позволяет вам выйти за пределы значений по умолчанию для обычных и административных пользователей. Изображение здесь отображает текущий вывод команды **semanage login -l.** Если вы не помните тонкостей команды **semanage**, возможно, будет проще использовать этот экран для сопоставления существующих пользователей с различными контекстами. Нажмите

Add, чтобы открыть окно **Add User Mapping**, показанное на **рисунке 4-18**. На этом рисунке также показано, как можно переклассифицировать пользователя с именем **michael** в тип пользователя SELinux **user_u.**

РИСУНОК 4-16 Логические значения в инструменте администрирования SELinux

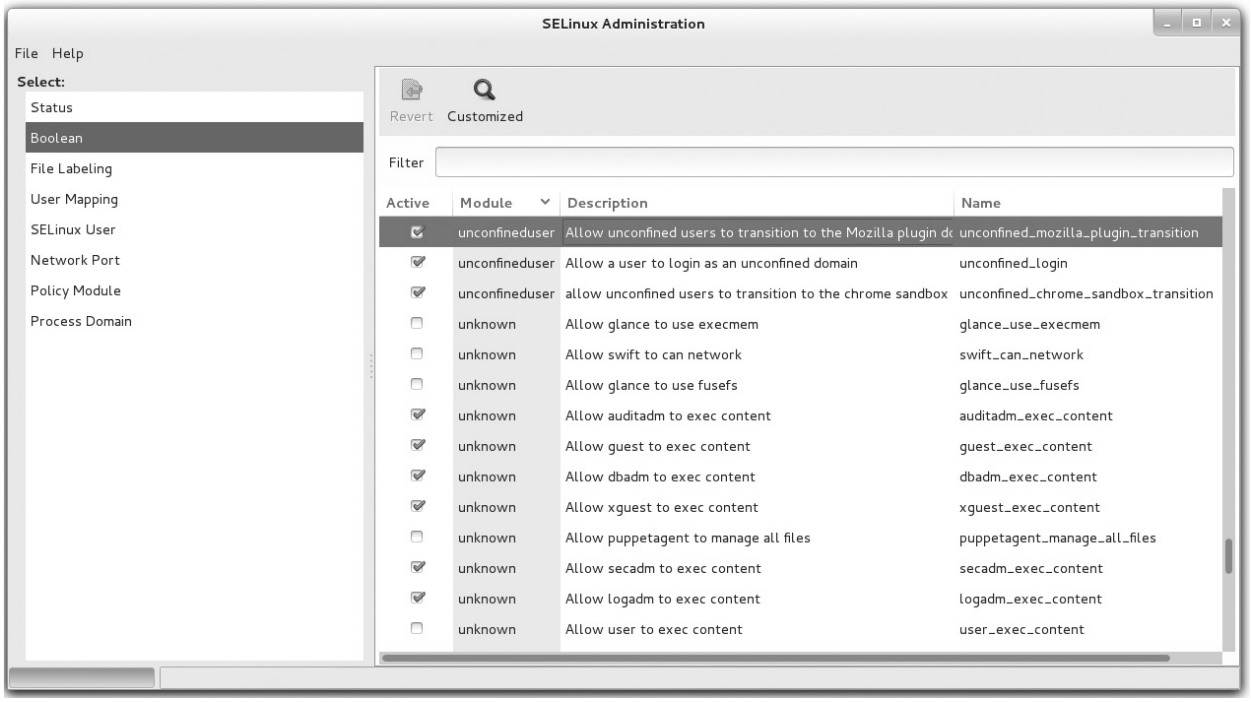
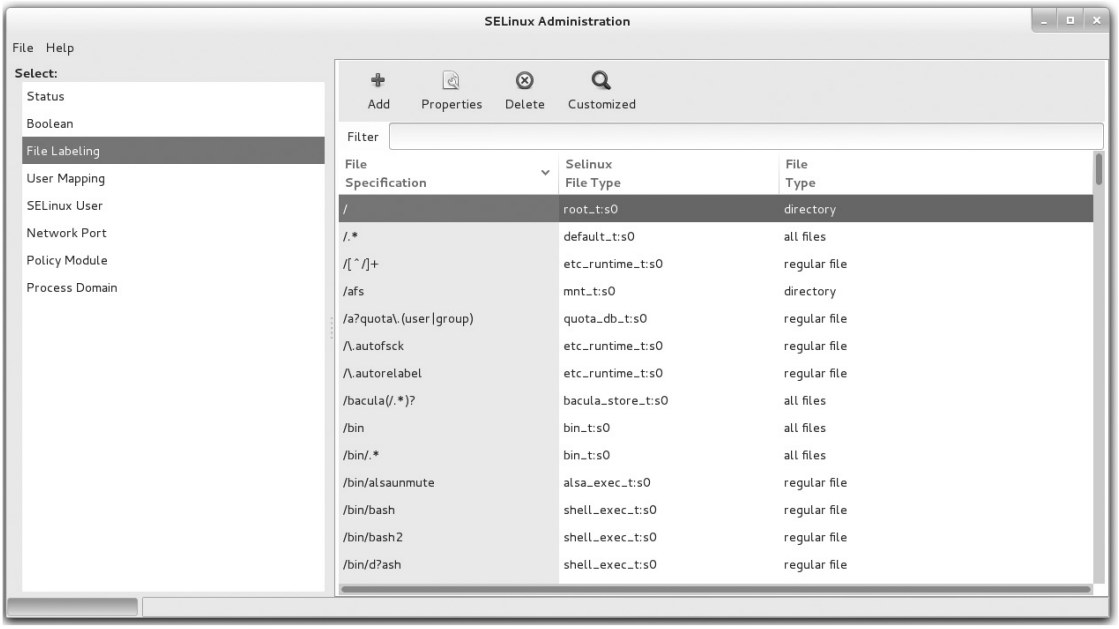


ТАБЛИЦА 4-12 Выбранные логические параметры SELinux

Логическое значение	Описание
fcron_crond	Поддерживает правила **fcron** для планирования заданий
cron_can_relabel	Позволяет заданиям **cron** изменять метку контекста файла SELinux
mount_anyfile	Разрешает использование команды **mount** для любого файла
daemons_use_tty	Позволяет сервисным демонам использовать терминалы по мере необходимости
daemons_dump_core	Поддерживает запись файлов ядра в корневой каталог верхнего уровня
virt_use_nfs	Поддерживает использование файловых систем **NFS** для виртуальных машин.
virt_use_comm	Поддерживает подключение виртуальных машин к последовательным и параллельным портам
virt_use_usb	Поддерживает использование USB-устройств для виртуальных машин
virt_use_samba	Поддерживает использование файловых систем **CIFS** (Common Internet File System) для виртуальных машин.
guest_exec_content	Позволяет пользователям guest_u выполнять скрипты
xguest_exec_content	Предоставляет пользователям xguest_u право выполнять сценарии
user_exec_content	Позволяет пользователям **user_u** выполнять скрипты
staff_exec_content	Позволяет пользователям **staff_u** выполнять скрипты
sysadm_exec_content	Позволяет пользователям **sysadm_u** выполнять скрипты

РИСУНОК 4-17. Типы файлов в инструменте администрирования SELinux



Пользователь SELinux

В разделе «Пользователь SELinux» (SELinux User) вы можете указать и изменить роли по умолчанию для стандартных пользователей, таких как обычные пользователи (**user_u**), системные пользователи (**system_u**) и неограниченные пользователи (**undefined_u**).

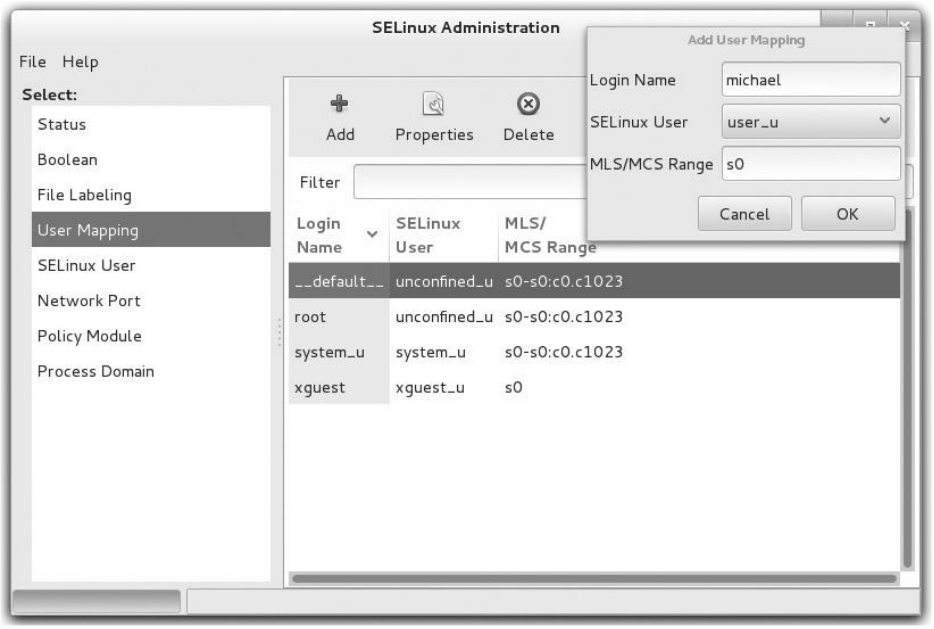
Сетевой порт

Раздел **Network Port** связывает стандартные порты со службами.

Модуль политики

В разделе «Модуль политики» указывается номер версии политики SELinux, применяемой к каждому модулю.

РИСУНОК 4-18. Сопоставьте пользователя в инструменте управления SELinux.



Домен процесса (Process Domain)

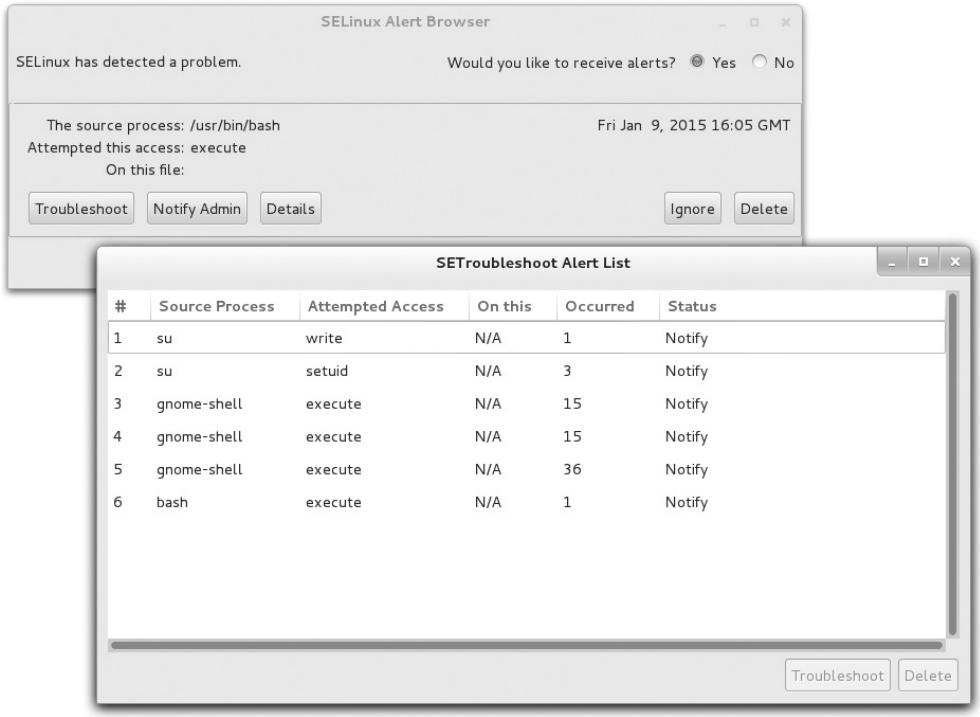
Домен процесса позволяет вам изменить состояние SELinux на режим **Permissive** или **Enforcing** для одного домена процесса, а не для всей системы.

Браузер устранения неполадок SELinux

RHEL 7 включает браузер устранения неполадок (Troubleshoot) SELinux, показанный на **рисунке 4-19.** В нем содержатся советы и рекомендации по любым проблемам, с которыми вы можете столкнуться, на языке, понятном администраторам Linux, часто включая команды, которые вы можете запустить и которые помогут решить данную проблему.

Чтобы запустить браузер с рабочего стола GNOME, нажмите **Приложения | Разное | SELinux** (**Applications | Sundry | SELinux**) Устраните неполадки или запустите **sealert -b** из командной строки на основе графического интерфейса. Команда доступна из пакета **setroubleshoot-server**.

РИСУНОК 4-19 Предупреждения безопасности с помощью браузера для устранения неполадок SELinux



УПРАЖНЕНИЕ 4-3
Проверьте тип пользователя SELinux

В этом упражнении вы настроите пользователя с типом SELinux **staff_u** и протестируете результаты. Вам потребуется графический интерфейс и хотя бы один обычный пользователь, отличный от пользователя root.

1. При необходимости создайте обычного пользователя. Даже если у вас уже есть обычный пользователь, второй обычный пользователь для этого упражнения может снизить риски. Пользователи всегда могут быть удалены, как обсуждалось в Главе 8. Для этого команды **useradd user1** и **passwd user1** создают пользователя с именем **user1** с паролем.

2. Просмотрите типы текущих пользователей SELinux с помощью команды **semanage login -l.**
3. Сконфигурируйте желаемого пользователя в качестве пользователя **staff_u** с помощью команды **semanage login -a -s staff_u user1**. Подставьте по желанию для **user1**.
4. Если вы полностью вошли в GUI, выйдите из системы. Нажмите Система | Выйдите (System | Log Out) из системы и нажмите «Выйти» в появившемся окне.
5. Войдите в GUI, используя новую учетную запись **staff_u, user1** (или что-то еще, что вы могли настроить на шаге 3). Если вы еще не видите экран входа в GUI, нажмите **alt-f1** или **alt-f7**.
6. Попробуйте различные административные команды. У вас есть доступ к команде su? Как насчет sudo? Вы можете прочитать это упражнение после прочтения главы 8, если вы не знаете, как использовать sudo. Какие административные инструменты, обсуждаемые до сих пор в этой книге, доступны? Есть ли разница, запускается ли этот инструмент из командной строки GUI или из меню GUI?
7. Выйдите из новой учетной записи **user1** и войдите в обычную учетную запись.
8. Удалить нового пользователя из списка **staff_u**; если это **user1**, вы можете сделать это с помощью команды **semanage login -d user1**.
9. Подтвердите восстановленную конфигурацию командой **semanage login -l**.

СЦЕНАРИИ И РЕШЕНИЯ	
Файл не может быть прочитан, записан или выполнен.	Проверьте текущее владение и разрешения с помощью команды **ls -l**. Примените изменения владельца с помощью команд **chown** и **chgrp**. Примените изменения прав с помощью команды **chmod**.
Доступ к защищенному файлу необходим для одного пользователя.	Настройте **ACL** с помощью команды **setfacl** для предоставления доступа.
Служба SSH недоступна на сервере.	Предполагая, что служба **SSH** работает (требование RHCE), убедитесь, что брандмауэр поддерживает SSH-доступ с помощью команды **firewall-cmd --list-all**; при необходимости измените его с помощью инструмента **firewall-config**.
Принудительный режим не установлен для SELinux.	Установите режим принудительного применения с помощью команды **setenforceforcing**. Проверьте настройки загрузки по умолчанию в **/etc/selinux/config.**
Необходимо восстановить контексты файлов SELinux по умолчанию в каталоге.	Примените команду **restorecon -F** к целевому каталогу. Используйте **ключ -R** для рекурсивного изменения контекста для всех файлов и подкаталогов.
Неожиданный сбой, когда SELinux установлен в принудительном режиме.	Используйте команду **sealert -a /var/log/audit/audit.log** или средство устранения неполадок SELinux, чтобы найти дополнительную информацию об ошибке; иногда предлагаемое решение включено.
Необходимо изменить параметры SELinux для пользователя.	Примените **команду setsebool -P** к соответствующему логическому значению.

РЕЗЮМЕ СЕРТИФИКАЦИИ

Эта глава была посвящена основам безопасности на уровне RHCSA. В любой системе Linux безопасность начинается с владения и прав, связанных с файлом. Право собственности может быть разделена на **пользователей, группы и другие** (**users, groups, and others**). Разрешения могут быть разделены на **чтение, запись и выполнение (read, write, and execute)** схема, известная как дискреционное управление доступом. Права доступа к файлам по умолчанию основаны на значение **umask** для пользователя. Разрешения могут быть расширены с помощью **SUID**, **SGID** и **закрепленных битов (sticky bits)**.

Списки управления доступом (**ACLs**) могут добавить другое измерение к дискреционным элементам управления доступом. Когда настроено на смонтированный том, списки **ACL** могут быть настроены для замены основных прав **ugo/rwx**. Списки управления доступом (**ACL**s) может быть включен в общие каталоги NFSv4.

Межсетевые экраны могут предотвращать связь на всех портах, кроме нужных. Стандартные порты для большинство сервисов определены в файле **/etc/services**. Однако некоторые службы могут не использовать все протоколы, определенные в этом файле. Брандмауэр RHEL 7 по умолчанию поддерживает доступ только к локальным SSH сервер.

Команда **ssh-keygen** создает пары ключей, защищенные парольной фразой, которые можно использовать для аутентификации на SSH-сервере без передачи пароля пользователя по сети.

SELinux обеспечивает еще один уровень защиты, используя обязательный контроль доступа. С помощью множества доступных пользователей SELinux, объектов, типов файлов (**users, objects, file types**) и диапазонов MLS, элементы управления SELinux могут помочь убедиться, что нарушение в одной службе не приведет к проблемам с другими службами.

Две минуты тренировки.

Вот некоторые из ключевых моментов целей сертификации в Главе 4.

Основные права доступа к файлам

- Стандартные разрешения для файлов Linux - чтение, запись и выполнение, которые могут различаться пользователя, группы и другие пользователи.
- Специальные разрешения включают **SUID**, **SGID** и закрепленные биты **(sticky bits).**
- Права пользователя по умолчанию основаны на значении **umask**.
- Владение и права могут быть изменены с помощью команды **chown**, **chgrp** и **chmod**.
- Специальные атрибуты файла могут быть перечислены с помощью команды **lsattr** и изменены командой **chattr**.

Списки контроля доступа и многое другое

- **ACL** могут быть перечислены и изменены в файловых системах, смонтированных с опцией **acl**. В файловых системах **XFS** и **ext4**, созданных в RHEL 7, такая опция включена по умолчанию.
- Каждый файл уже имеет **ACL** на основе стандартного владения и прав доступа.
- Вы можете настроить **ACL** для файла, чтобы заменить стандартные владения и разрешения для указанных пользователей и групп в выбранных файлах. Фактические **ACL** могут зависеть от маски.

- Пользовательских списков управления доступом (**ACL**) для файла недостаточно; выбранным пользователям и группам также необходим доступ в каталоги, которые содержат такие файлы.
- Так же, как пользовательские списки ACL могут поддерживать специальный доступ для выбранных пользователей, они также могут запретить доступ к другим выбранным пользователям.
- **ACL** могут быть настроены в общих каталогах **NFS**.

Основное управление брандмауэром

- Стандартные брандмауэры Linux основаны на системе ядра **Netfilter** и на инструменте **iptables**.
- Стандартные брандмауэры Linux предполагают использование некоторых портов и протоколов, перечисленных в **/etc/services**.
- Стандартный брандмауэр RHEL 7 поддерживает удаленный доступ **к локальному SSH-серверу.**
- Брандмауэр RHEL 7 можно настроить с помощью инструмента настройки межсетевого экрана с графическим интерфейсом пользователя или консольный командой **firewall-cmd**.

Защита SSH с помощью аутентификации на основе ключей

- Команды настройки **SSH** включают **ssh-keygen** и **ssh-copy-id**.
- Домашние каталоги пользователей включают в себя собственный подкаталог **.ssh** файлов конфигурации, с закрытыми и открытыми ключами SSH, подходящие для парольных фраз.
- Пары **личного/открытого (Private/public)** ключей можно настроить с помощью парольных фраз с помощью команды **ssh-keygen**.
- Публичные ключи можно передавать в домашние каталоги пользователей на удаленных системах с помощью команда **ssh-copy-id**.

Учебник по Linux с улучшенной безопасностью (Security-Enhanced Linux)

- **SELinux** может быть настроен на принудительный, разрешительный или отключенный **(enforcing, permissive, or disabled)** режимом, с целенаправленной или MLS политикой, с помощью команды **setenforce**. Настройки загрузки по умолчанию хранится в файле **/etc/selinux/config**.
- Пользовательские опции для SELinux могут быть установлены командой **semanage login**.
- Метки SELinux содержат разные контексты, такие как пользователь, роли, типы и уровни MLS **(user, roles, types, and MLS levels)**.
- Логическими **(booleans)** значениями SELinux можно управлять с помощью команды **setsebool**; Для постоянного изменения требуется ключ **-P**.
- Контексты SELinux можно изменить с помощью команды **chcon** и восстановить по умолчанию с помощью команды **restorecon**.
- Команда **sealert** и браузер устранения неполадок SELinux могут использоваться для интерпретировать проблемы, описанные в файле **audit.log** в каталоге **/var/log/audit**.

САМОТЕСТИРОВАНИЕ

Следующие вопросы помогут вам оценить ваше понимание материала, представленного в этом глава. Поскольку на экзаменах Red Hat нет вопросов с несколькими вариантами ответов, нет вопросов с несколькими вариантами ответов

появляются в этой книге. Эти вопросы исключительно проверяют ваше понимание главы. Получать результаты, не запоминание пустяков, это то, что рассчитывает на экзамены Red Hat. Может быть более одного правильного ответа на многие из этих вопросов.

Основные права доступа к файлам

1. Какая команда настраивает права на чтение и запись для файла с именем question1 в локальном каталоге, для владельца файла, без прав доступа для любого другого пользователя?

   _____

2. Какая отдельная команда меняет владельца пользователя на профессора (**professor**) и владельца группы на помощников (**assistants**) для локального файла с именем **question2**?

   _____

3. Какая команда изменит атрибуты файла с именем **question3**, чтобы только вы могли добавить в этот файл?

   _____

Списки контроля доступа и многое другое

4. Какая команда читает текущие списки ACL для локального файла с именем **question4**? Предположим, что файл находится на файловой системе с поддержкой ACL.

   _____

5. Какая отдельная команда дает членам группы «менеджеры» (**managers**) доступ для чтения к проекту5 (**project5**) к файлу в каталоге **/home/project**? Предположим, что группа менеджеров уже имеет права для чтения и выполнения к этому каталогу.

   _____

6. Какая команда препятствует доступу членов группы с именем **temp** к файлу **secret6** в каталоге **/home/project**?

   _____

Основное в управление брандмауэром

7. Какой номер порта **TCP/**IP связан со службой **HTTP**?

   _____

8. Перечислите полную команду **firewall-cmd**, чтобы постоянно разрешать входящий **HTTP**-трафик по в зоне умолчанию (**default**) в **firewalld**.

   _____

Защита SSH с помощью аутентификации на основе ключей

9. Какая команда настраивает пару **секретный/открытый (private/public)** ключ с использованием **DSA**?

_____

10. В каком подкаталоге домашнего каталога пользователя находится файл **authorized_keys**?

_____

Учебник по Linux с улучшенной безопасностью

11. Какая команда настраивает SELinux в принудительном (**enforcing**) режиме?
    _____

12. Какая команда отображает состояние SELinux текущих пользователей?
    _____

13. Какая команда перечисляет все логические (**boolean**) настройки для SELinux?
    _____

LAB ВОПРОСЫ

Некоторые из этих лабораторий включают в себя упражнения по настройке. Вы должны делать эти упражнения только на тестовых машинах. Предполагается, что вы выполняете эти упражнения на виртуальных машинах, таких как KVM, и они не используется для производства.

Red Hat представляет свои экзамены в электронном виде. По этой причине большинство лабораторий в этом и будущем главы доступны из средств массовой информации, которая сопровождает книгу. Для лаборатории этой главы, посмотрите в подкаталоге Глава 4/. Если вы еще не настроили RHEL 7 в системе, обратитесь к Главе 1 для инструкции по установке.

Ответы для каждой лаборатории следуют за ответами самопроверки для вопросов, которые заполняются.

Лабораторная работа

Во время экзаменов Red Hat задания будут представлены в электронном виде. Таким образом, эта книга также представляет большинство лабораторий в электронном виде. Для получения дополнительной информации см. Раздел «Лабораторные вопросы» в конце главы 4.

Экзамены Red Hat уникальны тем, что полагаются на лабораторные работы и практические демонстрации. Имейте в виду, что хотя лабораторные занятия 5, 6 и 7 охватывают разные темы, они предназначены для последовательной работы.

Лабораторная работа 1

В этой лабораторной работе вы изучите роль бита **SUID**.

1. Удалите разрешения **SUID** для исполняемого файла **/usr/bin/passwd** с помощью команды **chmod u-s /usr/bin/passwd**.
2. Попробуйте запустить команду **passwd** как обычный пользователь. Что происходит? Ваш пароль изменился? Попробуйте еще раз. Что сработало при запросе текущего пароля?
3. Вернитесь к учетной записи пользователя **root** и восстановите разрешения **SUID** для файла **/usr/bin/passwd**.
4. Попробуйте снова запустить команду **passwd**, как обычный пользователь. Изменить пароль. Что происходит в этот раз?

Лабораторная работа 2

В этой лабораторной работе вы создадите сценарий, настроите разрешения для файла для этого сценария, а затем сконфигурируете **ACL** для этого сценария, который будет выполняться обычным пользователем.

1. В текстовом редакторе откройте файл script1 в каталоге / usr / local / bin.
2. Введите следующие строки в этом файле:

   #!/bin/bash
   /bin/ls > filelist

3. Сохраните файл.
4. Попробуйте выполнить этот скрипт от имени пользователя **root**. Что происходит?
5. Настройте разрешения на выполнение для пользователя-владельца файла **script1** с помощью команды **chmod u + x /usr local/bin/script1**. Можете ли вы сейчас выполнить скрипт от имени пользователя **root**?
6. Измените разрешения для файла **script1**, созданного в лаборатории 1, с помощью команды **chmod 700 /usr/local/bin/script1**.
7. Войдите в систему как обычный пользователь. Попробуйте выполнить этот скрипт. Что происходит?
8. По умолчанию поддержка **ACL** уже включена в файловых системах **XFS**. Настройте **ACL** для чтения и выполнения для одного обычного пользователя в файле **script1**. Проверьте с помощью команды **getfacl**.
9. Повторите шаг 7, войдя в систему как обычный пользователь, которому предоставлены привилегии **ACL** для скрипта **script1**. Что происходит?
10. Если вы хотите восстановить исходную конфигурацию, удалите файл **script1** из каталога **/usr/local/bin**.

Лабораторная работа 3

В этой лабораторной работе вы установите **ACL** для обычного пользователя в домашнем каталоге пользователя **root, /root**. Начните с настройки **ACL** для каталога, а затем просмотрите результаты из учетной записи обычного пользователя. Какие файлы можно прочитать из каталога **/root**? Что еще нужно сделать, чтобы настроить **ACL** для определенного файла в каталоге **/root**?

Просто убедитесь, что отключили ACL в / root каталоге, когда лаборатория будет завершена.

Лабораторная работа 4

В этой лабораторной работе вы рассмотрите процесс отключения и повторного включения SELinux в системе. Просмотрите текущее состояние SELinux с помощью команды **sestatus**. Вы можете отключить SELinux через файл **/etc/selinux/config** или с помощью инструмента администрирования SELinux. Сделайте это и перезагрузите систему. Попробуйте команду **sestatus** снова. Повторно включите SELinux и перезагрузите систему. Что происходит? Процесс занимает много времени? Сколько раз система перезагружается? Что бы произошло, если бы вам пришлось ждать перезапуска и процесса перезагрузки во время экзамена Red Hat?

Лабораторная работа 5

В этой лабораторной работе вы настроите одного обычного пользователя в категории **SELinux guest_u**. Помните, что соответствующие команды начинаются с **semanage login**. При наличии параметров с пользователем **__default__** существует несколько способов удовлетворить требования этой лабораторной работы.

Прежде чем вносить какие-либо изменения, запишите текущее состояние пользователей SELinux; один метод с помощью следующей команды, которая записывает вывод в файл **selinuxusers.txt**:

 # semanage login -l> selinuxusers.txt

Ваша работа будет продолжена в **Лабораторной работе 6**

.

Лабораторная работа 6

Теперь с обычным пользователем в категории **guest_u** посмотрите, что вы можете сделать. Попробуйте следующие действия:

1. Попробуйте войти в GUI. Что происходит?
2. Войдите в обычную консоль. Попробуй **ssh** на другую машину. Что происходит?
3. Вернувшись в консоль, попробуйте команду **su -** и введите **пароль root**. Что происходит?
4. Попробуйте некоторые команды **system-config-***. Что происходит?

Лабораторная работа 7

В этой лабораторной работе вы деактивируете логическое (**boulean**) значение **guest_exec_content SELinux**. Сделайте это с помощью команды, которая гарантирует, что изменение сохраняется после перезагрузки. После завершения выйдите из системы в качестве настроенного гостевого пользователя, а затем снова войдите в систему. Скопируйте двоичный файл, такой, как **/bin/ls**, в домашний каталог пользователя и запустите программу:

$ cp /bin/ls ~
$ ~/ls

Что происходит? Попробуйте запустить программу из каталога **/tmp**.
Когда процесс завершится, войдите в GUI как неограниченный пользователь и просмотрите пользователей SELinux в инструменте администрирования SELinux с графическим интерфейсом. (Для этого допустимо входить в GUI с учетной записью администратора **root**.) Используйте раздел сопоставления пользователей, а также инструменты, доступные для восстановления исходной конфигурации, как описано в файле **selinuxusers**. Не забудьте деактивировать логическое значение **guest_exec_content**.

Лабораторная работа 8

В этой лабораторной работе вы создадите новый каталог **/ftp** с контекстами SELinux, подходящими для этого каталога. Он должен основываться на контекстах в каталоге **/var/ftp/pub**. Используйте знания, которые вы получили в этой главе, чтобы завершить эту лабораторную работу. Когда вы закончите, восстановите исходные контексты в каталоге **/ftp**. Чем отличаются контексты SELinux? Из какого файла пришли восстановленные контексты? Являются ли восстановленные контексты такими же, как при создании каталога **/ftp**?

Лабораторная работа 9

На этом этапе у вас должны быть некоторые данные, доступные в файле **audit.log** в каталоге **/var/log/audit**. Если это так, попробуйте прочитать файл журнала. Если нет,

убедитесь, что SELinux установлен в принудительном режиме (или, по крайней мере, в разрешающем режиме). Убедитесь, что служба аудита работает с командой **systemctl status auditd**.

Примените команду **sealert -a** к этому файлу; Вы можете перенаправить этот вывод в текстовый файл для облегчения просмотра. Можете ли вы определить проблемы в файле? Какие пользователи были перечислены в этом файле? Можете ли вы идентифицировать пользователей и группы по их UID и GID номерам? Для получения дополнительной информации о UID и GID см. Главу 8. Есть ли какие-либо предлагаемые решения?

ОТВЕТЫ НА САМОПРОВЕРКУ

Основные права доступа к файлам

1. Команда, которая настраивает разрешения на чтение и запись для файла с именем **question1** в локальном каталоге, без каких-либо разрешений для любого другого пользователя, является

   # chmod 600 question1

2. Единственная команда, которая меняет владельца пользователя на профессор (**professor**) и владельца группы на помощников (**assistants**) для отмеченного файла

   # chown professor.assistants question2

   допустимо заменить двоеточие (:) точкой (.).

3. Команда, которая изменяет атрибуты файла с именем **question3**, чтобы позволить вам только добавлять к этому файлу

   # chattr +a question3

Списки контроля доступа и многое другое

4. Команда, которая читает текущие списки **ACL** для локального файла с именем **question4**

   # getfacl question4

5. Единственная команда, которая дает членам группы с именем «менеджеры» (**manager**) доступ на чтение к файлу **project5** в каталоге **/home/project:**

   # setfacl -m g:manager:r /home/project/project5

6. Команда, которая препятствует доступу членов группы с именем **temps** к файлу **secret6** в каталоге **/home/project:**

   # setfacl -m g:temps:- /home/project/secret6

Основное управление брандмауэром

7. Номер порта **TCP/IP**, связанный со службой **HTTP**, равен **80**.
8. Команда **firewall-cmd**, которая постоянно разрешает входящий HTTP-трафик в зоне по умолчанию в **firewalld:**

 # firewall-cmd --permanent --add-service=http

Защита SSH с помощью аутентификации на основе ключей

9. Команда **ssh-keygen -t dsa**.
10. Каждый пользователь с открытыми ключами, хранящимися в файле **author_keys**, может найти этот файл в подкаталоге **.ssh/** его домашнего каталога.

Учебник по Linux с улучшенной безопасностью

11. Команда, которая настраивает SELinux в принудительном (**enforcing**) режиме:

    # setenforce enforcing

12. Команда, которая отображает состояние SELinux текущих пользователей:

    # semanage login -l

13. Команда, которая перечисляет все логические настройки для SELinux:

    # semanage boolean -l

Ответы Лабораторной работы

Лабораторная работа 1

Лабораторная работа 1 предназначена для того, чтобы вы могли практиковаться в настройке разрешений, связанных с битом **SUID /usr/bin/passwd**.

Лабораторная работа 2

Лабораторная работа 2 демонстрирует подход к созданию сценария, принадлежащего пользователю, исполняемого другим пользователем. Если сценарий правильно выполняется обычным пользователем, настроенным в ACL, вы найдете файл с именем **filelist** в локальном каталоге.

Лабораторная работа 3

Конфигурирование списков **ACL** в административном каталоге **/root** является плохой практикой безопасности. Тем не мение, это отличный способ проиллюстрировать возможности списков **ACL** в системе и узнать, как он может выбранные обычные пользователи для внутренних «святилищ» корневого административного аккаунта. Из-за рисков, отключите ACL, когда лаборатория будет завершена. Если выбранный пользователь **michael**, один их методов сделать это выполнить следующую команду:

 # setfacl -b u:michael /root

Лабораторная работа 4

Эта лаборатория предназначена для повышения осведомленности о времени и усилиях, необходимых для отключения и повторного включения SELinux в принудительный режим. Если вы переключаетесь между отключенным и разрешающим режимами, то необходимое время и усилия должны быть примерно таким же. Если вам

придется перенастроить SELinux в принудительном режиме, вы можете потерять драгоценное время во время экзамена Red Hat, потому что больше ничего нельзя сделать, пока система перезагружается и перемаркируется.

Лабораторная работа 5

Обычные пользователи в RHEL 7 работают с типом (types) пользователей SELinux **unconfined_u**. Таким образом, есть несколько ограничений на их учетные записи пользователей. Если инструкции на экзамене или из корпоративной политики требуют определенных ограничений на обычных пользователях вы можете настроить пользователя **__default__** с типом пользователя **SELinux user_u**.

В качестве альтернативы, если вам сказали настроить конкретных пользователей на ограниченный тип, такой как **xguest_u** или **staff_u**, может потребоваться выполнить несколько команд **semanage login**. Если вам нужно пересмотреть синтаксис введите команду **semanage login**, запустите **man semanage-login**.

Лабораторная работа 6

После тестирования пользователя как пользователя guest_u большинство администраторов захотят, чтобы у обычных пользователей было больше привилегии. Однако **пользователь guest_u** подходит для таких систем, как пограничный сервер, где вы хотите что бы учетные записи пользователей должны быть заблокированы.

Лабораторная работа 7

Пользователи, настроенные с типом пользователя **SELinux guest_u**, обычно могут выполнять сценарии даже в их собственные домашние каталоги. Это может измениться с логическим значением **guest_exec_content**, описанным в лабораторных условиях. Успех в этой лабораторной работе основан на простом сравнении: может ли двоичный файл выполняться с и без активных логических значений.

Хотя самый простой способ восстановить исходную конфигурацию - это управление с помощью графического интерфейса SELinux, вы также должны знать, как использовать такие команды, как следующие, который отключает пользовательский Тип пользователя SELinux для пользователя **michael**:

 # semanage login -d michael

Лабораторная работа 8

Успех в этой лаборатории можно измерить сначала командой **ls -Zd**. При применении к **/ftp** и каталогу **/var/ftp/pub** должны привести к тому же списку ролей, объектов, типов и MLS (**roles, objects, types, and ML**) SELinux варианты для каждого каталога.

Затем выполните команду **restorecon -R / ftp** и проверьте еще раз тип SELinux каталога **/ftp**.

Если он изменился, это означает, что вы пропустили команду **semanage fcontext** для изменения файла по умолчанию контекста, описанные в главе.

Лабораторная работа 9

Каждый будет экспериментировать с SELinux по-разному. Так что результаты этой лаборатории до вас. Цель состоит в том, чтобы проанализировать текущий соответствующий файл журнала и обработать его в командной строке. Попробуйте определить проблемы, связанные с каждым предупреждением. Хотя вы не сможете

решить многие проблемы SELinux, по крайней мере до второй половины этой книги, вы должны быть в состоянии определить проблемы или, по крайней мере, пользователей и/или команды, связанные с каждым предупреждением.