Глава 11

Системные сервисы и SELinux

ЦЕЛИ СЕРТИФИКАЦИИ

11.01 Конфигурация системы Red Hat
11.02 Linux с улучшенной безопасностью
11.03 Secure Shell Server
11.04 Контрольный список безопасности и конфигурации
✓ Двухминутная тренировка
Q & A Самопроверка

Эта глава посвящена общим задачам, которые вы будете выполнять на работе. Эти задачи относятся к подробной настройке служб уровня **RHCE**.
RHEL 7 включает базовые файлы конфигурации системы в каталог **/etc/sysconfig**, вызываемый различными службами и заданиями **cron**. Неотъемлемой частью этого подхода является настройка **SELinux**, так как он включает в себя значительное количество пользовательских опций для различных сервисов.

Вы протестируете эти инструменты на одной службе, которую вы можете установить на всех системах бастионных систем: **SSH**. Поскольку это общий сервис для всех таких систем, хакеры «черной шляпы» хотят найти слабость в **SSH**. Поэтому в этой главе описывается, как сделать службы SSH более безопасными. Это первая глава, в которой вы будете использовать три виртуальные машины, созданные в главах 1 и 2.

В этой главе вы также сконфигурируете логические параметры, используемые **SELinux** для защиты различных сервисов. Хотя **SELinux** является распространенным источником разочарований, с ним легче справиться, если вы знаете опции, которые поддерживают желаемые функции.

Кроме того, в этой главе описана основная процедура, обеспечивающая работоспособность различных служб, доступ к ним из удаленных систем и запуск при следующей перезагрузке системы.

ВНУТРИ ЭКЗАМЕНА

Внутри раздела содержатся задачи, которые будут повторяться в оставшейся части книги:

- Установите пакеты, необходимые для предоставления услуги.

При установке файлового сервера **Samba** или **DNS-сервера** только для кэширования имен вы будете использовать те же инструменты. Да, это те же команды **rpm и yum**, а также инструменты управления пакетами, **описанные в главе 7**. Чтобы сэкономить время, вы можете использовать эти команды для установки служб, описанных в главах с 12 по 17.

- Настройка службы для запуска при система загружается.
- Настройка **SELinux** для поддержки службы.
- Настройка маркировки портов **SELinux**, чтобы службы могли использовать нестандартные порты.

В то время как подробная настройка отдельных служб является предметом рассмотрения каждой главы, необходимо выполнить шаги, необходимые для настройки службы. во время процесса загрузки основаны на общих командах, таких как **systemctl**. Кроме того,

конфигурация **SELinux** для поддержки службы требует доступа и настройки аналогичных параметров. Как предлагается во введении, особое внимание уделяется службе SSH.

- Настройка аутентификации на основе ключей

Требование к ключам экзамены **RHCSA и RHCE**, описанные **в главе 4**. Возможно, вы захотите ознакомиться с разделом этой главы под названием «**Защита SSH с помощью аутентификации на основе ключей**». Учитывая важность безопасности **SSH**, в этой главе мы также рассмотрим следующее задачи:

- Настройка дополнительных параметров, описанных в документации

ЦЕЛЬ СЕРТИФИКАЦИИ 11.01

Конфигурация системы Red Hat

В этом разделе вы ознакомитесь с основной информацией о том, как настраиваются службы в системах **Red Hat**. Фактический процесс, связанный со службой, является демоном. Такие демоны являются исполняемыми файлами, обычно хранящимися в каталоге **/usr/sbin. Red Hat** настраивает пользовательские параметры и многое другое в каталоге **/etc/sysconfig**. На эти файлы ссылаются задания **cron** или системные модули.

Управление Сервисом

Как обсуждалось на протяжении всей книги, службы управляются файлами конфигурации системного модуля обслуживания. Как описано в **Главе 4**, вы можете использовать **systemctl** для запуска, остановки или перезапуска службы. Во многих случаях вы можете использовать **systemctl** для перезагрузки сервиса с измененными файлами конфигурации, не отключая пользователей, подключенных в данный момент.

Хотя настоящие демоны находятся в каталогах **/usr/sbin**, системные файлы **systemd** делают больше. Они вызывают демонов с параметрами, настроенными в их файлах модулей в каталоге **/lib/systemd/system**. Затем файлы модуля ссылаются на файлы конфигурации, специфичные для службы.

RHEL 7 поддерживает совместимость с традиционной системой сценариев инициализации, которая была обнаружена в более ранних версиях **Red Hat Enterprise Linux**. Скрипты инициализации старого стиля все еще находятся в Каталог **/etc/rc.d/init.d**, на который ссылаются символические ссылки в подкаталогах **/etc/rc.d/rcX.d**. А старая служебная команда в каталоге **/usr/sbin** является оболочкой для команды **systemctl**. Другими словами, следующие команды функционально идентичны:

```
# systemctl restart sshd
# service sshd restart
```

Системные сервисы

Файлы в каталоге **/etc/sysconfig** обычно используются с заданиями **cron** и системными модулями. Они так же разнообразны, как и файлы конфигурации модуля, включенные в каталог **/lib/systemd/system**. Поскольку они включают базовые параметры конфигурации для каждого демона, они управляют основными операциями каждой службы.

В большинстве случаев каждый из этих файлов поддерживает использование переключателей, как описано в соответствующих справочных страницах. Например, файл **/etc/sysconfig/httpd** можно использовать для настройки пользовательских параметров запуска **веб-сервера Apache**. В этом файле директива **OPTIONS** передает ключи демону **/usr/sbin/httpd**, как определено на странице руководства **httpd**.

Широкая иллюстрация конфигурационных процессов

В общем, когда вы настраиваете сетевую службу в **Linux**, выполните общие шаги, описанные в этом разделе. Фактические шаги, которые вы предпринимаете, могут отличаться; Например, вы можете сначала изменить параметры **SELinux**. Иногда вам нужно протестировать службу локально и удаленно, прежде чем убедиться, что служба запускается автоматически при следующей перезагрузке.

1. Установите сервис с помощью команды, такой как rpm или yum. В некоторых случаях вам может понадобиться установить дополнительные пакеты.
2. Отредактируйте соответствующие файлы конфигурации службы. Обычно вам нужно изменить и настроить несколько файлов конфигурации, например, для почтового сервера **Postfix** в каталоге **/etc/postfix**.
3. Модифицируйте логические выражения **SELinux**. Как будет обсуждаться далее в этой главе, большинство служб имеют более одного логического значения **SELinux**. Например, вы можете изменить различные логические значения **SELinux**, чтобы файловый сервер **Samba** мог обмениваться файлами в режиме **чтения/записи** или только для чтения.
4. Запустите сервис. Вам также необходимо убедиться, что служба запускается при следующей загрузке системы, как будет описано далее в этой главе.
5. Протестируйте сервис локально. Убедитесь, что он работает с соответствующего клиента (ов) и в локальной системе.
6. Установите соответствующие политики брандмауэра, основываясь на **firewalld**, **TCP Wrappers** и специфичных для службы файлах конфигурации. Настройте доступ к нужным пользователям и системам.
7. Протестируйте сервис удаленно. Если нужные порты открыты, служба должна работать так же, как и при локальном подключении. При правильных ограничениях услуга не должна быть доступна нежелательным пользователям или системам.

Доступные инструменты конфигурации

В целом, наиболее эффективно настраивать различные службы из командной строки. Администратор, который знает службу, может настроить основные операции всего за несколько минут. Однако большинство администраторов не могут специализироваться ни на чем. С этой целью **Red Hat** разработала ряд инструментов для настройки. При правильном использовании эти инструменты изменят правильные файлы конфигурации. Некоторые устанавливаются с каждым сервисом; другие должны быть установлены отдельно. Большинство этих инструментов доступны из интерфейса командной строки **GUI** с помощью команды **system-config- *.**

Инструменты, используемые в этой книге (и еще несколько), приведены в **Таблице 11-1**.

ТАБЛИЦА 11-1 Инструменты настройки Red Hat

Инструменты	Команды	Назначение
Add/Remove Software	**gpk-application**	Оконный менеджер команды **yum;** управляет текущей конфигурацией программного обеспечения
Authentication Configuration	**authconfig*,system-config-authentication**	Конфигурация баз данных **user/group** и аутентификация клиента
Date/Time Properties	**system-config-date**	Управление текущим часовым поясом, клиент **NTP**

Firewall Configuration	**firewall-config**	Настройка брандмауэров на основе **firewalld**, маскировка и переадресация **ip адресов**.
Language Selection	**system-config-language**	Выбор языка в **GUI**
Network Connections	**nm-connection-editor**	Подробный инструмент настройки сетевого устройства
Network Management	**nmtui**	Настройка сетевого устройства/DNS-клиента в консоли
Printer Configuration	**system-config-printer**	Управление сервером печати **CUPS**
SELinux Management	**system-config-selinux**	Конфигурация логических выражений **SELinux**, меток, пользователей и т. д.
Software Update	**gpk-update-viewer**	Используется для просмотра и установки доступных обновлений для установленных пакетов
User Manager	**system-config-users**	Управление и настройка пользователей и групп

ЦЕЛЬ СЕРТИФИКАЦИИ 11.02

Linux с улучшенной безопасностью

Security-Enhanced Linux (SELinux) обеспечивает еще один уровень безопасности. Разработанный Агентством национальной безопасности США, **SELinux** усложняет хакерам «черной шляпы» использование или доступ к файлам или службам, даже в скомпрометированных системах. **SELinux** назначает **контекст** каждому объекту, например файлу, устройству или сетевому сокету. Контекст объекта сообщает, какие действия может выполнять процесс (или субъект в жаргоне **SELinux**).

Основные параметры **SELinux** были рассмотрены в **главе 4**, поскольку это также является требованием для сертификации RHCSA. Для RHCE фокус **SELinux** относится к различным сервисам. В частности, вам нужно знать, как настроить **SELinux** для поддержки веб-сервера **Apache**, служба системы доменных имен (**DNS**), система управления базами данных **MariaDB**, файловый сервер **Samba**, служба **SMTP**, служба **Secure Shell (SSH**) и служба протокола сетевого времени (**NTP**).

Требования к каждой из этих услуг рассматриваются в этой и последующих главах этой книги. Поскольку конфигурация **SELinux** для каждой службы требует использования одних и тех же команд и инструментов, они описаны здесь.

Ключевыми командами и инструментами, обсуждаемыми в этом разделе, являются **getsebool, setsebool, chcon, restorecon, ls -Z** и инструмент управления **SELinux**. Хотя это те же инструменты, которые использовались в главе 4, основное внимание уделяется другому. Чтобы проверить, команды **getsebool** и **setsebool** устанавливают логические параметры в файлах каталога **/sys/fs/selinux/booleans**. Логическое значение - это двоичная опция 1 или 0, которая соответствует да или нет.

Каталоги размещения флагов(логических переменных) SELinux

При настройке **SELinux** для службы вы обычно вносите изменения в логические параметры в виртуальной файловой системе **/sys**. Посмотрите на файлы в каталоге **/sys/fs/selinux/booleans**. Имена файлов носят описательный характер.

Например, логическое значение **http_enable_homedirs** разрешает или запрещает доступ к домашним каталогам пользователей через сервер Apache. Это отключено по умолчанию. Другими словами, если вы настроили сервер **Apache в главе 14** для обслуживания файлов из домашних каталогов пользователей без изменений в **SELinux**, веб-сервер не сможет получить доступ к файлам.

Подобные проблемы являются распространенным источником разочарования для администраторов систем **RHEL**. Они выполняют всю работу по настройке службы, проверяют конфигурацию, проверяют документацию, думают, что все сделали правильно, и, тем не менее, служба не работает так, как им хочется. Решение состоит в том, чтобы сделать **SELinux** частью того, что вы делаете для настройки службы.

В качестве примера выполните следующую команду:

$ cat /sys/fs/selinux/booleans/httpd_enable_homedirs

0 0

Это два нуля. Предположительно, один логический параметр предназначен для текущей настройки, а другой - для постоянной настройки. На практике числа не отражают различий, по крайней мере, для RHEL 7, но различия все еще существуют. Из-за этой проблемы лучший способ увидеть текущее состояние логического выражения - это команда **getsebool**. Например, команда

$ getsebool httpd_enable_homedirs

приводит к следующему выводу:

httpd_enable_homedirs --> off

В итоге, если текущий параметр равен 0, следующая команда будет активировать **httpd_enable_homedirs** только до перезагрузки системы:

 # setsebool httpd_enable_homedirs 1

Повторим из главы 4 способ сделать изменение постоянным из командной строки с помощью команды **setsebool -P**, в данном случае:

 # setsebool -P httpd_enable_homedirs 1

!EXAM!
Многие логические значения SELinux, относящиеся к сервису, описаны в локальной документации; для просмотра списка связанных man-страниц выполните команду man -k selinux.
!!!!!!!!!!

Сервисные категории SELinux Booleans

В каталоге **/sys/fs/selinux/booleans** находится около 300 псевдофайлов. Поскольку имена файлов в этом каталоге носят описательный характер, вы можете использовать команды фильтра базы данных, такие как **grep**, чтобы помочь классифицировать эти логические значения. Основываясь на некоторых службах, обсуждаемых в этой книге, следующие подходящие команды фильтрации:

$ ls /sys/fs/selinux/booleans | grep http
$ ls /sys/fs/selinux/booleans | grep samba

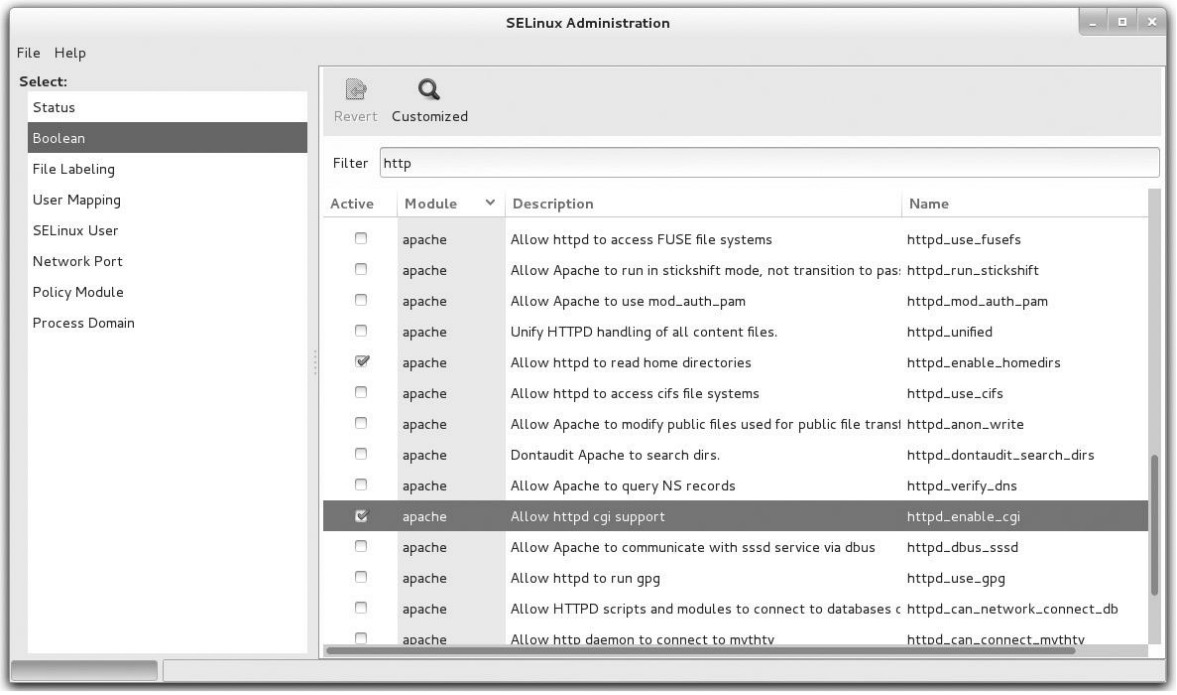
$ ls /sys/fs/selinux/booleans | grep nfs

Вскоре вы познакомитесь с каждой из этих категорий логических значений более подробно. Для краткого описания доступных логических значений с их текущим состоянием выполните команду **semanage boolean -l**. Команда **semanage** является частью пакета **policycoreutils-python**.

Конфигурация логических переменных(флагов) с помощью инструмента управления SELinux

Одним из преимуществ инструментов с графическим интерфейсом пользователя является представление о «большой картине». С помощью инструмента управления **SELinux** вы можете просматривать активные логические значения и быстро получать представление о том, настроен ли **SELinux** на использование нескольких или более параметров, связанных со службой. Как уже говорилось **в главе 4**, вы можете запустить инструмент управления **SELinux** в среде рабочего стола с графическим интерфейсом с помощью команды system-config-selinux. На левой панели щелкните **Boolean**. Это открывает доступ к группа логических значений в правой части окна. Обратите внимание на **http**-фильтр, добавленный на **рис. 11-1.** Он фильтрует систему для всех логических значений, связанных с веб-сервером Apache.

РИСУНОК 11-1 Фильтрация логических значений с помощью инструмента управления SELinux.



Сравните список с выводом команды **ls /sys/fs/selinux/booleans | grep http**, описанная ранее. Обратите внимание на различия. На самом деле вы увидите больше связанных с **Apache** логических значений в инструменте **GUI**, поскольку фильтр в инструменте управления **SELinux** фильтрует логические имена и описания **SELinux**.

Ряд категорий показан на левой панели окна инструмента управления **SELinux**; они описаны в следующих разделах. Основное внимание здесь будет уделено логической категории, где большинство политик **SELinux** настроены.

В некоторых случаях логическое значение связано с требованием к контексту файла **SELinux**. Например, логическое значение **httpd_anon_write** работает только в том случае, если связанные файлы и каталоги помечены типом **public_content_rw_t**. Чтобы установить этот тип,

скажем, в каталог **/var/www/html/files (и подкаталоги),** вы должны выполнить следующую команду:

 # chcon -R -t public_content_rw_t /var/www/html/files

Настройки логических переменных (флагов)

Настройки **логических переменных (флагов),** обсуждаемые в следующих разделах, делятся на несколько категорий. Они основаны на сервисах, определенных в целях **RHCE**. Настройки **SELinux** выполняются не в одном месте. Например, если вы включите логическое значение **httpd_enable_homedirs**, вам все равно придется настроить файл **/etc/httpd/conf.d/userdir.conf** для поддержки доступа к домашним каталогам пользователей. Только после того, как **SELinux** и **Apache** настроены с такой поддержкой, пользователи могут подключаться к своим домашним каталогам через этот сервер **Apache**.

Поскольку в настоящее время нет булевых значений **SELinux**, связанных со службой протокола сетевого времени (**NTP**), в этом обсуждении нет отдельного раздела для логических переменных (флагов) **NTP**.

Обычные и безопасные HTTP-сервисы

Существует ряд директив **SELinux**, которые помогают защитить **веб-сервер Apache**, как показано в следующем списке. Большинство из них просты и говорят сами за себя. Они упорядочены по имени логического файла, как показано в каталоге **/sys/fs/selinux/booleans**. Хотя эти логические значения могут применяться к другим веб-серверам, Red Hat предполагает использование **веб-сервера Apache**. В описаниях указывается конфигурация, если логическое значение активно.

- **httpd_anon_write** Позволяет веб-службе записывать в файлы, помеченные с типом **public_content_rw_t**.
- **httpd_builtin_scripting** Разрешает доступ к сценариям, обычно связанным с **PHP**. Включено по умолчанию.
- **httpd_can_check_spam** Поддерживает использование **SpamAssassin** для веб-приложений электронной почты.
- **httpd_can_network_connect** Разрешает **скриптам и модулям Apache** доступ к внешним системам по сети; обычно отключается, чтобы минимизировать риски для других систем.
- **httpd_can_network_connect_cobbler** Позволяет **скриптам и модулям Apache** получать доступ к внешнему серверу установки **Cobbler**. Если вам не нужно подключаться ни к каким службам, кроме **Cobbler**, вам следует отключить логическое значение **httpd_can_network_connect**.
- **httpd_can_network_connect_db** Разрешает **соединения с портами сервера базы данных**; более конкретный, чем **httpd_can_network_connect**.
- **httpd_can_network_memcache** Включает доступ к серверу **memcache** по сети.
- **httpd_can_network_relay** Поддерживает использование службы **HTTP** в качестве прямого или обратного прокси.
- **httpd_can_sendmail** Позволяет **Apache** отправлять электронные письма.
- **httpd_dbus_avahi** Поддерживает доступ **к сервису avahi** через систему сообщений **D-bus**. По умолчанию отключено.
- **httpd_enable_cgi** Позволяет запускать сценарии **Common Gateway Interface (CGI).** Включено по умолчанию; требует, чтобы скрипты были помечены с типом файла **httpd_sys_script_exec_t**.
- **httpd_enable_ftp_server** Позволяет **Apache** прослушивать **порт FTP (обычно 21)** и работать как **FTP-сервер**.

- **httpd_enable_homedirs** Позволяет **Apache** обслуживать контент из домашних каталогов пользователей через директиву **UserDir**.
- **httpd_execmem** Поддерживает программы, такие как написанные на **Java или Mono**, для которых требуются адреса памяти, которые являются исполняемыми и записываемыми.
- **httpd_mod_auth_ntlm_winbind** Разрешает доступ к базам данных аутентификации **Microsoft NT LAN Manager (NTLM) и Winbind**; требует установленного и активного модуля **mod_auth_ntlm_winbind для Apache**.
- **httpd_mod_auth_pam** Поддерживает доступ **PAM** для аутентификации пользователя; требует установленного и активного модуля **mod_auth_pam для Apache**.
- **httpd_read_user_content** Позволяет **веб-серверу Apache** читать все файлы в домашних каталогах пользователей.
- **httpd_setrlimit** Позволяет изменять ограничения **дескриптора файла Apache**.
- **httpd_ssi_exec** Поддерживает исполняемые включения на стороне **сервера (SSI)**.
- **httpd_sys_script_anon_write** Позволяет сценариям **HTTP** записывать файлы, помеченные как **public_content_rw_t**.
- **httpd_tmp_exec** Позволяет **Apache** запускать исполняемые файлы из каталога **/tmp**.
- **httpd_tty_comm** Поддерживает доступ к терминалу; необходим **Apache** для запроса пароля, если закрытый ключ сертификата **TLS** защищен паролем.
- **httpd_unified** Включает доступ ко всем файлам с пометкой **httpd _ * _ t**, независимо от того, доступны ли они только для чтения, доступны для записи или являются исполняемыми. По умолчанию отключено.
- **httpd_use_cifs** Поддерживает доступ из **Apache** к общим файлам и каталогам **Samba**, помеченным с типом файла **cifs_t**.
- **httpd_use_fuse** Поддерживает доступ из **Apache** к файловым системам **FUSE**, таким как тома **GlusterFS**.
- **httpd_use_gpg** Позволяет **Apache** использовать **GPG** для шифрования.
- **httpd_use_nfs** Поддерживает доступ из **Apache к общим NFS-файлам и каталогам**, помеченным с типом файла **nfs_t**.
- **httpd_use_openstack** Позволяет **Apache** получать доступ к портам **OpenStack**.

Сервис Службы Имён

Демон службы имен (**named**) основан на программном обеспечении **Berkeley Internet Name Domain (BIND),** которое является службой **DNS** по умолчанию **RHEL 7**. Если вы поддерживаете авторитетную зону **DNS**, активируйте логическое значение **named_write_master_zones**. Затем локальное программное обеспечение **DNS** может перезаписать файлы мастер-зоны.

В целом, этот раздел не относится к **RHCE**, поскольку в целях указано, что все, что вам нужно сделать с **DNS**, - это настроить сервер имен только для кэширования. Такие серверы не являются полномочными для конкретного домена. Следовательно, указанный DNS-логический параметр не применяется, поскольку такие **DNS**-серверы не имеют файлов главной зоны.

RHEL включает в себя **Unbound DNS resolver**, небольшую службу, которую вы можете установить вместо **BIND** для предоставления сервера имен кэширования.

MariaDB

Два логических значения **SELinux** связаны исключительно со службой базы данных **MariaDB**. Как правило, вам не нужно менять значения по умолчанию.

- **mysql_connect_any** Позволяет **MariaDB/MySQL** подключаться ко всем портам. По умолчанию отключено.

- **selinuxuser_mysql_connect_enabled** Позволяет пользователям **SELinux** подключаться к локальному серверу **MariaDB/MySQL** с помощью доменного сокета **Unix**. По умолчанию отключено.

NFS

Некоторые из базовых логических выражений **SELinux**, связанных с серверами сетевой файловой системы (**NFS**), включены по умолчанию, что позволяет вам совместно использовать каталоги с сервером **NFS**.

- **nfs_export_all_ro** Позволяет экспортировать общие каталоги **NFS** с разрешениями только для чтения. Включено по умолчанию.
- **nfs_export_all_rw** Позволяет экспортировать общие каталоги **NFS** с разрешениями на **чтение/запись**. Включено по умолчанию.
- **use_nfs_home_dirs** Поддерживает доступ к домашним каталогам из удаленных систем **NFS**. По умолчанию отключено.
- **virt_use_nfs** Включает доступ виртуальных гостей к смонтированным файловым системам **NFS**.

Samba

Логические значения **Samba** обычно не включены по умолчанию. Таким образом, в большинстве конфигураций вам нужно активировать один или несколько логических выражений **SELinux**, чтобы соответствовать изменениям в файлах конфигурации **Samba**. Эти логические значения включают следующее:

- **samba_create_home_dirs** Позволяет **Samba** создавать новые домашние каталоги, например, для пользователей, которые подключаются из других систем, обычно через модуль **PAM pam_mkhomedir.so.**
- **samba_domain_controller** Включает настройку локального сервера **Samba** в качестве локального контроллера домена в сети в стиле **Microsoft Windows**.
- **samba_enable_home_dirs** Поддерживает совместное использование домашних каталогов пользователей.
- **samba_export_all_ro** Разрешает общий доступ к файлам и каталогам в режиме только для чтения.
- **samba_export_all_rw** Позволяет совместно использовать файлы и каталоги в режиме чтения/записи.
- **samba_run_unconfined** Позволяет **Samba** запускать неограниченные сценарии, хранящиеся в каталоге **/var/lib/samba/scripts**.
- **samba_share_fusefs** Поддерживает совместное использование файловых систем, смонтированных под файловыми системами **FUSE (fusefs)**.
- **samba_share_nfs** Поддерживает совместное использование файловых систем, смонтированных под **NFS**.
- **smbd_anon_write** Позволяет **Samba** изменять файлы в общедоступных каталогах, настроенных с использованием контекстов **SELinux public_content_rw_t и public_content_r_t.**
- **use_samba_home_dirs** Поддерживает использование удаленного сервера **Samba** для локальных домашних каталогов.
- **virt_use_samba** Позволяет виртуальным машинам использовать файлы, общие для **Samba**.

SMTP

Оба логических значения **SELinux**, связанные со службами **SMTP**, работают по умолчанию с **сервером Postfix**. Логические значения **httpd_can_sendmail** был описан ранее. Другое логическое значение **Postfix** включено по умолчанию:

- **postfix_local_write_mail_spool** Позволяет **Postfix** записывать в локальные каталоги спула.

SSH

Логические значения **SELinux**, связанные с **SSH-соединениями**, перечислены далее. Все отключены по умолчанию:

- **ssh_chroot_rw_homedirs** Позволяет службе **SSH** с поддержкой **chroot** читать и записывать файлы из домашних каталогов пользователей.
- **allow_ssh_keysign** Позволяет аутентификацию на основе хоста; не требует имени пользователя или публичной/частной аутентификации на основе парольной фразы.
- **ssh_sysadm_login** Поддерживает доступ пользователей, настроенных с ролью **sysadm_r**. Это не включает пользователя с правами администратора; в общем, более безопасно входить в систему как обычный пользователь, соединяясь с парольными фразами, прежде чем проходить аутентификацию с правами администратора.

Контексты файлов SELinux

Изменения, сделанные с помощью команды **chcon**, не являются постоянными. Хотя они переживают перезагрузку, они не переживают перемаркировку. Отключение **SELinux** системы может произойти, когда **SELinux** отключен, а затем снова включен. Команда **restorecon** перемаркирует целевой каталог. Сконфигурированные контексты **SELinux** хранятся в каталоге **/etc/selinux/target/contexts/files.**

Версия этого каталога по умолчанию включает в себя три важных файла:

file_contexts Базовые контексты файла для всей системы
file_contexts.homedirs Файловый контекст для каталога **/home** и всех его подкаталогов
media для съемных устройств, которые могут быть подключены после установки

Если вам нужно изменить контексты файловой системы, чтобы пережить перемаркировку, команда **semanage** может помочь. Например, если вам нужно настроить каталог **/www** для виртуальных веб-сайтов, следующая команда гарантирует, что контексты файлов будут соответствавать этому каталогу (и подкаталогам) даже после перемаркировки:

 # semanage fcontext -a -t httpd_sys_content_t "/www(/.*)?"

Указанная команда добавляет правило контекста файла в файл **file_contexts.local** в Каталог **/etc/selinux/target/contexts/files**. Для обсуждения значения регулярного выражения **(/.*)?** , обратитесь к главе 4.

В то время как команда **semanage** управляет различными политиками **SELinux**, здесь основное внимание уделяется контекстам файлов, которые представлены параметром **fcontext**. Доступные переключатели команд описаны **в таблице 11-2**.

ТАБЛИЦА 11-2 Командные переключатели для semanage fcontext

Switch	Description
-a	**Add (**Добавить**)**
-d	**Delete (**Удалить**)**

-D	**Delete all (**Удалить всё**)**
-f	**File type (**Тип файла**)**
-l	**List (**Список**)**
-m	**Modify (**Изменить**)**
-n	**No heading (**Нет заголовка**)**
-r	**Range (**диапазон**)**
-s	**SELinux user name (used for user roles)(** Имя пользователя SELinux (используется для пользовательских ролей**)**
-t	**SELinux file type (**Тип файла SELinux**)**

Маркировка портов SELinux

Политика **SELinux** контролирует каждое действие, которое процесс может выполнить над определенным объектом, таким как файл, устройство или сетевой сокет. Открытие сокета **TCP** и прослушивание сетевого порта - это одно из тех действий, которые вы можете контролировать и ограничивать с помощью политики **SELinux**.

Если одна из служб, описанных в предыдущем разделе, настроена на прослушивание нестандартного порта, по умолчанию целевая политика **SELinux** будет запрещать это действие. Фактически, **SELinux** использует метки для управления не только доступом к файлам или устройствам, но также и сетевым портам.

Вы можете получить список всех меток портов **SELinux**, выполнив команду **semanage**:

 # semanage port -l

Фильтрация для определенной строки может помочь определить, какие порты службе разрешено прослушивать. Как показано в следующем примере, служба SSH ограничена прослушиванием порта 22:

 # semanage port -l | grep ssh
ssh_port_t tcp 22

Аналогично, метка **http_port_t** регулирует порты, которые **Apache** может прослушивать, тогда как **http_cache_port_t** идентифицирует порты, разрешенные веб-прокси:

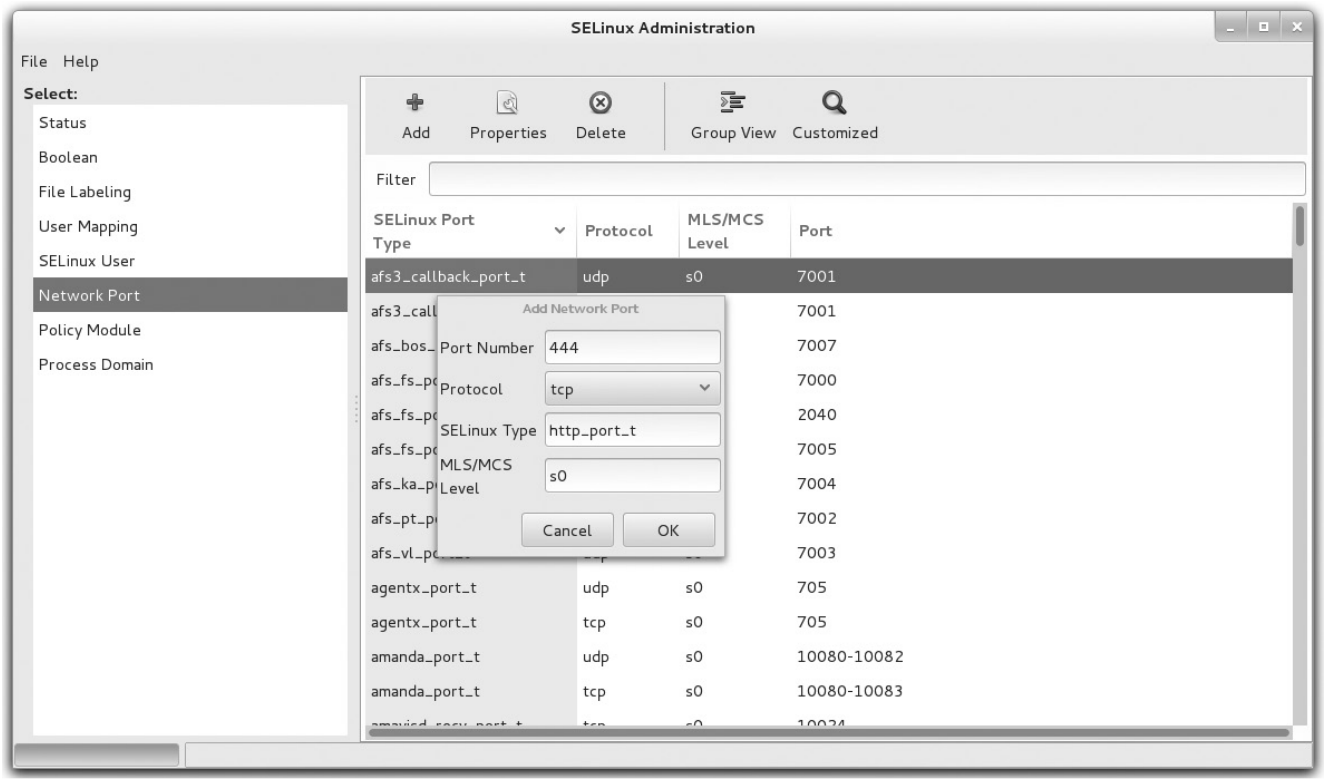
 # semanage port -l | grep http
http_cache_port_t tcp 8080, 8118, 8123, 10001-10010
http_cache_port_T udp 3130
http_port_t tcp 80, 81, 443, 488, 8008, 8009, 8443, 9000

Если вам нужно изменить метку, чтобы служба могла прослушивать нестандартный порт, используйте команду **semanage**. В следующем примере политика **SELinux** модифицируется, чтобы позволить **Apache** прослушивать порт **444**:

 # semanage port -a -t http_port_t -p tcp 444

Излишне говорить, что вы можете достичь того же результата с помощью инструмента управления **SELinux**, как показано на **рисунке 11-2**.

Рис. 11-2 добавление сетевого порта с помощью средства управления SELinux



УПРАЖНЕНИЕ 11-1

Настройте новый каталог с соответствующими контекстами SELinux

В этом упражнении вы создадите новый каталог **/ftp** с контекстами **SELinux**, которые соответствуют стандартному каталогу для **FTP-серверов**. Это упражнение демонстрирует, как это делается с помощью команды **chcon**, а также с эффектами команд **restorecon** и **semanage**.

1. Создайте каталог **/ftp**. Используйте команду **ls -Zd /ftp**, чтобы определить контексты **SELinux** в этом каталоге. Сравните это с контекстами в каталоге **/var/ftp**
2. Измените контексты в каталоге **/ftp**, чтобы они соответствовали контекстам в каталоге **/var/ftp**. Наиболее эффективный метод - следующая команда:

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

   Хотя **ключ -R** не требуется, мы включили его, чтобы помочь вам привыкнуть к идее рекурсивного изменения контекста.
3. Запустите команду **ls -Zd /ftp**, чтобы просмотреть измененные контексты в этом каталоге. Теперь он должен соответствовать контекстам в каталоге **/var/ftp**.
4. Запустите следующую команду, чтобы увидеть, что происходит, когда **SELinux** перемаркируется:

   # restorecon -Rv /ftp

   Что эта команда сделала с контекстами каталога **/ftp**?
5. Чтобы сделать изменения в каталоге **/ftp** постоянными, вам нужна помощь от команды **semanage** с параметром **fcontext**. Поскольку аналога командному переключателю **chcon --reference** нет, следующая команда указывает роль пользователя и тип файла на основе настроек по умолчанию для каталога **/var/ftp**:

   # semanage fcontext -a -s system_u -t public_content_t "/ftp(/.*)?"

6. Просмотрите результаты. Во-первых, команда **semanage** не изменяет текущие контексты **SELinux** каталога **/ftp**. Далее просмотрите содержимое **file_contexts.local** в каталоге **/etc/selinux/target/contexts/files**. Он должен отражать только что выполненную команду **semanage**.
7. Перезапустите команду **restorecon** из шага 4. Изменяет ли она контексты **SELinux** каталога **/ftp** сейчас?

ЦЕЛЬ СЕРТИФИКАЦИИ 11.03

Secure Shell Server

Red Hat Enterprise Linux устанавливает пакеты сервера **Secure Shell (SSH)** по умолчанию, используя **RPM-серверы openssh, openssh-client и openssh**. Глава 2 была посвящена клиентским программам **SSH**, включая **ssh, scp и sftp**, а в главе 4 мы обсуждали, как обеспечить доступ **SSH** с помощью **аутентификации на основе ключей**. Основное внимание в этом разделе уделяется серверу **SSH**. Безопасный демон **sshd** прослушивает весь входящий трафик через **TCP-порт 22**. Файлы конфигурации сервера **SSH** находятся в каталоге **/etc/ssh.**

Файлы конфигурации сервера SSH

Файлы конфигурации сервера **SSH** хранятся в каталоге **/etc/ssh**. Функциональность этих файлов обобщена здесь:

- **moduli** Поддерживает метод обмена ключами **группы Диффи-Хеллмана** с простыми числами и генераторами случайных ключей
- **ssh_config** Включает конфигурацию для локального клиента **SSH, обсужденную в Главе 2**
- **sshd_config** Определяет конфигурацию сервера **SSH**, которая подробно обсуждается далее в этой главе.
- **ssh_host_ecdsa_key** Включает закрытый ключ хоста для локальной системы на основе **алгоритма ECDSA**
- **ssh_host_ecdsa_key.pub** Включает открытый ключ хоста для локальной системы на основе **алгоритма ECDSA**
- **ssh_host_rsa_key** Включает закрытый ключ хоста для локальной системы на основе алгоритма **RSA**
- **ssh_host_rsa_key.pub** Включает открытый ключ хоста для локальной системы на основе **алгоритма RSA**

Настройте сервер SSH

Вам не нужно много делать, чтобы настроить сервер SSH для основной работы. Установите пакеты, описанные ранее, активируйте службу и убедитесь, что она активна при следующей перезагрузке системы. Как уже говорилось в главе 1, стандартный **SSH-порт (TCP 22**) открыт в брандмауэре **RHEL 7** по умолчанию.

Однако цели **RHCE** указывают, что вы должны быть готовы «настроить дополнительные параметры, описанные в документации». Из-за общего характера этой задачи в этом разделе будут рассмотрены все активные и закомментированные параметры в версии по умолчанию файла конфигурации сервера **SSH**.

Файл конфигурации сервера **SSH** - это **/etc/ssh/sshd_config**. Команды в комментариях, как правило, по умолчанию. Поэтому, если вы хотите установить нестандартный порт для службы **SSH**, вы можете изменить закомментированную директиву

 # Port 22

что-то вроде этого:

Port 2222

Предполагая, что брандмауэр и **SELinux** разрешают доступ через этот порт, вы сможете подключиться из удаленной системы с помощью **команды ssh -p 2222 server1.example.com**. Если сервер **SSH** имеет другое имя, замените **server1.example.com** на нужное.

Хотя следующая закомментированная строка (**#AddressFamily any**) подразумевает, что сервер **SSH** использует адреса как **IPv4**, так и **IPv6**, можно ограничить доступ к одному из этих типов адресов с помощью ключевых слов **inet** и **inet6**, которые соответствуют **IPv4** и **IPv6** соответственно:

AddressFamily inet
AddressFamily inet6

Значение по умолчанию, показанное с помощью следующих директив **ListenAddress**, заключается в том, чтобы прослушивать соединения SSH на всех локальных адресах **IPv4 и IPv6:**

#ListenAddress 0.0.0.0
#ListenAddress ::

Вы можете ограничить **SSH** прослушиванием **IPv4** или **IPv6**-адресов определенных сетевых карт. Это может помочь ограничить доступ к серверу **SSH** для определенных сетей.

Следующая закомментированная директива настраивает версию **SSH**. Как отмечалось ранее, **версия 1 SSH** считается небезопасной. **Версия 2 используется по умолчанию**:

#Protocol 2

Поскольку **SSH версии 1 отключен**, вам не нужно активировать следующую директиву, которая устанавливает ключ хоста для версии 1:

#HostKey /etc/ssh/ssh_host_key

Стандартные ключи **RSA и ECDSA** описаны в следующих строках. **ECDSA (Elliptic Curve DSA)** считается более безопасным, чем стандартный протокол **DSA**. Как правило, нет причин менять выбранные по умолчанию значения:

HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key

Следующие комментируемые директивы относятся к **эфемерному ключу SSH версии 1**. Такой ключ сервера будет обновляться каждый час с 1024 битами, но это все равно будет небезопасным.

#KeyRegenerationInterval 1h
#ServerKeyBits 1024

Следующая строка указывает, как часто ключ сеанса пересматривается. По умолчанию происходит повторное согласование после того, как объем данных шифра был передан по умолчанию («по умолчанию»), без ограничений по времени («нет»).

#RekeyLimit default none

В следующих строках первая не закомментированная директива отправляет все сообщения журнала в соответствующее средство ведения журнала. На основании конфигурации файла **/etc/rsyslog.conf** все сообщения, связанные со средством **AUTHPRIV**, записываются в файл **/var/log/secure**. Уровень информации – **INFO** и выше.

#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO

Чтобы ограничить атаки типа «отказ в обслуживании» (DOS), значение LoginGraceTime по умолчанию, показанное здесь, составляет две минуты. Другими словами, если процесс входа в систему не был завершен за это время, сервер SSH автоматически отключается от удаленного клиента.

#LoginGraceTime 2m

Директива, которая следует за документами, которые пользователь **root** может войти в систему, используя **SSH**:

#PermitRootLogin yes

Прямой вход в систему через SSH может быть небезопасным. Если вы настроили аутентификацию на основе секретного пароля на основе **закрытого/открытого** ключа из учетной записи администратора на портативном компьютере, это риск. Хакер «черной шляпы», который овладевает этой портативной системой, может затем подключиться к удаленному серверу с правами администратора. По этой причине обычно рекомендуется изменить эту директиву следующим образом:

PermitRootLogin no

Администраторы, которые входят в систему как обычные пользователи, могут использовать команду **su** или **sudo** в зависимости от ситуации, чтобы получить права администратора с меньшими рисками. Но если это не является обязательным требованием при сдаче экзамена, не вносите это изменение. На самом деле, это можно считать ошибкой на экзамене.

Далее, более безопасно сохранить следующую директиву, особенно в отношении закрытых и открытых ключей:

#StrictModes yes

Эта директива проверяет, установлены ли соответствующие разрешения для домашнего каталога пользователя и ключей **SSH**, перед авторизацией входа в систему.

Как отмечено в следующей директиве, по умолчанию количество попыток аутентификации на **соединение равно шести**. Вы можете уменьшить это число для дополнительной безопасности, но недостатком является то, что вы можете получить больше ложных срабатываний в журналах, связанных с законными пользователями, которые неправильно набрали свой пароль:

#MaxAuthTries 6

Следующая директива предполагает, что вы можете открыть до 10 сессий SSH для соединения:

#MaxSesssions 10

Следующая директива используется только **с SSH версии 1**. Надеемся, что вы не активировали эту версию **SSH**.

#RSAAuthentication yes

С другой стороны, следующая директива крайне важна, если вы хотите настроить аутентификацию на основе закрытого / открытого ключа в стандартном протоколе **SSH версии 2:**

#PubkeyAuthentication yes

Следующая директива подтверждает использование файла **author_keys** в системе для указания открытых ключей, которые можно использовать для аутентификации:

#AuthorizedKeysFile .ssh/authorized_keys

Следующая директива применяется только тогда, когда центр сертификации используется в процессе аутентификации:
#AuthorizedPrincipalsFile none

Следующие две директивы обычно игнорируются:

#AuthorizedKeysCommand none
#AuthorizedKeysCommandRunAs nobody

Следующая директива **Rhosts**, как правило, не используется, поскольку она применяется к **SSH версии 1 и менее защищенной Remote Shell (RSH)**:

#RhostsRSAAuthentication no

Хотя следующая директива может поддерживать использование файла /**etc/hosts.equiv** для ограничения числа подключаемых хостов, это обычно не рекомендуется. Тем не менее, это один из методов обеспечения безопасности на основе хоста **SSH**, выходящий за рамки возможного с такой альтернативой, как **TCP Wrappers**, как обсуждалось в главе 10.

#HostbasedAuthentication no

Как описано в **Главе 4**, файл **.ssh/known_hosts** хранит открытые ключи из удаленных систем и читается из-за следующего значения по умолчанию:

#IgnoreUserKnownHosts no

Следующая директива может помочь администраторам, которые конвертируют из **RSH** в **SSH**, поскольку они используют файлы **.rhosts и .shosts**. Однако, поскольку он не используется по умолчанию, целесообразно использовать следующую опцию:

#IgnoreRhosts yes

Для систем и пользователей, где **частные/публичные** парольные фразы не используются, необходима аутентификация на основе пароля, как это включено по умолчанию:

#PasswordAuthentication yes

В общем, вы никогда не должны разрешать пустые пароли из-за угроз безопасности:

#PermitEmptyPasswords no

Аутентификация по запросу-ответу обычно связана с одноразовыми паролями, общими для удаленных терминалов. Хотя он также может работать с **PAM**, он обычно отключен в SSH:

ChallengeResponseAuthentication no

Если вы настроили систему **Kerberos** для локальной сети с использованием **SSH** версии **1**, вы бы использовали некоторые из следующих опций. Первые два говорят сами за себя, так как они могут включить проверку Kerberos пользователя и настроить альтернативную проверку подлинности **Kerberos** или локальный пароль.

#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

В **версии 2 SSH** для аутентификации **Kerberos** используется библиотека универсального интерфейса прикладных программ служб безопасности (**GSSAPI**):

GSSAPIAuthentication = yes

Следующая директива уничтожает учетные данные **GSSAPI** при выходе из системы:

GSSAPICleanupCredentials = yes

Обычно проверки имени хоста являются строгими:
GSSAPIStrictAcceptorCheck = yes

Также возможен обмен ключами **GSSAPI**:

GSSAPIKeyExchange = yes

Аутентификация через модули **PAM** поддерживается:

UsePAM yes

При следующей настройке команду **ssh-agent** можно использовать для пересылки закрытых ключей в другие удаленные системы:

#AllowAgentForwarding yes

Со следующей строкой конфигурации соединения **TCP** могут быть переадресованы через соединение **SSH**:

#AllowTCPForwarding yes

Директива **GatewayPorts** обычно отключена, чтобы не позволять удаленным хостам подключаться к перенаправленным портам:

#GatewayPorts no

Следующая директива важна для всех, кому нужен удаленный доступ к инструменту с графическим интерфейсом через **X forwarding**:

X11Forwarding yes

Например, когда вы работаете из удаленного местоположения, вы можете подключиться и открыть инструменты **GUI** из вашей системы Red Hat дома или в нашем офисе через **SSH**, используя команду, подобную следующей:

 # ssh -X michael@Maui.example.com

Следующая директива помогает избежать конфликтов между локальным и удаленным графическим интерфейсом. Значение по умолчанию должно быть адекватным, если не используется более 10 дисплеев X11.

 # X11DisplayOffset 10

Обычно не требуется никаких изменений для следующего значения по умолчанию, связанного с тем, как отображение графического интерфейса привязано к серверу **SSH**:

 # X11UseLocalhost yes

Когда пользователи **SSH** входят в систему удаленно, следующий параметр означает, что они видят содержимое файл **/etc/motd**. Возможны разные сообщения, основанные на скрипте **cron**, настроенном в главе 9.

#PrintMotd yes

Это одна полезная настройка для администраторов, поскольку она документирует дату и время последнего входа в указанную систему:

#PrintLastLog yes

Директива **TCPKeepAlive** разрешает сообщения поддержки активности TCP, чтобы избежать зависания сеанса навсегда, если сетевое соединение, сервер **SSH** или любой подключенный клиент **SSH** не работает:

#TCPKeepAlive yes

Как правило, вы не должны включать эту опцию, потому что она несовместима с **X11Forwarding**:

#UseLogin no

Разделение привилегий, связанное со следующей директивой, устанавливает отдельный процесс после успешной аутентификации с привилегиями аутентифицированного пользователя:

UsePrivilegeSeparation sandbox

Следующая директива не отменяет по умолчанию **AuthorizedKeysFile** настройки в начале файла:

#PermitUserEnvironment no

Сжатие часто помогает ускорить обмен данными через соединение SSH. По умолчанию сжатие задерживается до тех пор, пока пароль не будет принят или пара секретного/открытого ключей не будет сопоставлена для аутентификации пользователя:

#Compression delayed

Иногда важно, чтобы сервер SSH удостоверился, что пользователь все еще хочет передавать данные. Это как клиенты отключены от чувствительных систем, таких как банковские счета. Но для административного соединения следующая опция отключает такие проверки:

#ClientAliveInterval 0

Если для **ClientAliveInterval** задано какое-то число, следующая директива указывает количество сообщений, которые могут быть отправлены до автоматического отключения этого клиента:

#ClientAliveCountMax 3

Следующая опция для уровня исправления применяется только к SSH версии 1:

#ShowPatchLevel no

Чтобы минимизировать риски подделки, следующая директива проверяет имена удаленных хостов на DNS-сервере или в файле **/etc/hosts**:

#UseDNS yes

Перечисленный здесь файл **PID** содержит идентификационный номер процесса запущенного процесса сервера **SSH**:

#PidFile /var/run/sshd.pid

Когда хакер «черной шляпы» пытается взломать **SSH**-сервер, он может попытаться установить несколько соединений, пытаясь одновременно войти в систему. Следующая директива ограничивает количество не аутентифицированных соединений, с которыми будет работать SSH-сервер. Для сервера **SSH** в административной системе это то, что вы могли бы уменьшить.

#MaxStartups 10

Следующая директива, если она активирована, будет поддерживать пересылку устройств:

#PermitTunnel no

Следующая директива может показаться хорошей идеей, но ее трудно реализовать на практике. Любой указанный каталог должен содержать все команды и файлы конфигурации в этом дереве каталогов, поскольку сеанс **SSH** будет привязан к указанному каталогу:

#ChrootDirectory none

Следующие директивы могут быть использованы для указания дополнительного текста для добавления к баннеру протокола **SSH** и для установки баннера по умолчанию:

#VersionAddendum none
 # Banner none

Следующие директивы позволяют клиенту устанавливать несколько переменных среды. Подробности обычно тривиальны между двумя системами Red Hat Enterprise Linux:

AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY
LC_MESSAGES
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE
AcceptEnv XMODIFIERS

Последняя директива поддерживает использование шифрования **SSH** для передачи файлов **SFTP**:

Subsystem sftp /usr/libexec/openssh/sftp-server

УПРАЖНЕНИЕ 11-2

Запустите сервер SSH на нестандартном порту

В этом упражнении вы настроите сервер **OpenSSH** для прослушивания **TCP-порта 2222**. Для достижения этой цели вам придется изменить не только политику SELinux, но также службу **SSH** и конфигурацию брандмауэра.

1. Посмотрите, какие порты службе **SSH** разрешено слушать, основываясь на текущей политике **SELinux**:

   # semanage port -l | grep ssh

2. Выполните следующую команду, чтобы разрешить **OpenSSH** прослушивать **TCP-порт 2222**:

   # semanage port -a -t ssh_port_t -p tcp 2222

3. Откройте файл **/etc/ssh/sshd_config** и измените строку

   # Port 22
   на
   Port 2222

4. Не забудьте разрешить **TCP-порт 2222** через зону по умолчанию брандмауэра, как показано здесь:

   # firewall-cmd --permanent --add-port 2222/tcp
   # firewall-cmd --reload

5. Перезагрузите службу **SSH**, чтобы применить изменения:

   # systemctl reload sshd

6. Если вы успешно выполнили предыдущие шаги, вы сможете войти в систему с удаленной системы, выполнив команду, подобную следующей:

$ ssh -p 2222 alex@192.168.122.50

7. Наконец, восстановите исходную настройку конфигурации **SSH**.

Пользовательская безопасность для SSH

Пользовательская безопасность может быть настроена в файле **/etc/ssh/sshd_config**. Для этого нам нравится добавлять директивы, которые ограничивают пользователей, которым разрешен доступ к системе через **SSH**. Ключ - это директива **AllowUsers**. Вы можете ограничить пользователем с помощью директивы, такой как

AllowUsers michael donna

В качестве альтернативы вы можете ограничить доступ каждого пользователя с определенных хостов с помощью следующей директивы, которая объединяет аспекты безопасности как на уровне пользователя, так и на уровне хоста:

AllowUsers michael@192.168.122.50 donna@192.168.122.150

Имейте в виду, что если запрос на доступ поступает из удаленной сети, маскирующийся межсетевой экран может назначить **IP-адрес маршрутизатора** удаленной системе. В этом случае вы не сможете заблокировать одну систему в удаленной сети.

Вы можете включить несколько связанных директив в файл **/etc/ssh/sshd_config**: **AllowGroups, DenyUsers** и **DenyGroups**.

Если вы хотите ограничить доступ к **SSH** для очень немногих пользователей, директива **AllowUsers** является самым простым решением. Для первой только что показанной директивы **AllowUsers** к этому **SSH-серверу** могут подключаться только пользователи **michael** и **donna**. Соответствующая директива **DenyUsers** или **DenyGroups** не требуется. Даже пользователь **root** не может подключиться через **SSH** при таких обстоятельствах.

Хотя сервер **SSH** запрашивает пароль у других пользователей, доступ запрещен, даже если удаленный пользователь вводит правильный пароль. Файл журнала **/var/log/secure** будет отражать это с помощью сообщения, подобного следующему:

User alex from 192.168.122.150 not allowed because not listed in AllowUsers

Host-Based Security для SSH

Хотя существуют способы настройки безопасности на основе хоста через файлы конфигурации **SSH**, этот процесс сложен. Это требует изменений как для серверов, так и для клиентов, и включает риски, которые, по нашему мнению, не являются необходимыми. Также можно настроить безопасность на основе хоста через локальный брандмауэр на основе зоны **firewalld**.

Самый простой метод обеспечения безопасности **SSH** на основе хоста основан на **TCP Wrappers**, как обсуждалось в **главе 10**. Для целей этой главы мы включили следующую директиву **в /etc/hosts.allow**, которая принимает соединения **SSH** от указанных сетевые адреса:

sshd : 127.192.168.122.

Чтобы убедиться, что доступ ограничен системами в указанных сетях, вы также должны включить следующую строку в **/etc/hosts.deny**:

sshd : ALL

Конечно, было бы более безопасно **включить ALL: ALL в /etc/hosts.deny,** но это может заблокировать связь с законными службами, которые вы настроили. Кроме того, другие порты уже должны быть защищены соответствующим правилом брандмауэра. Так что это может быть вариант, чтобы избежать во время экзамена Red Hat.

ЦЕЛЬ СЕРТИФИКАЦИИ 11.04

Контрольный список безопасности и конфигурации

Несколько шагов, необходимых для установки, настройки и защиты службы, повторяются. Поэтому мы суммируем их в этом разделе. При желании вы можете использовать этот раздел для подготовки к главам с 12 по 17. Он поможет вам установить необходимые службы, а также убедиться, что эти службы активны и доступны через брандмауэр, настроенный с соответствующими открытыми портами.

Установка серверных сервисов

Цели **RHCE** напрямую касаются восьми различных услуг. В этом разделе рассматриваются некоторые из различных способов установки этих служб. Если вы читали Главу 7, это должно быть в основном пересмотрено, но это также даст вам возможность подготовить систему, такую как виртуальная машина **server1.example.com**, для тестирования в главах с 12 по 17.

В этом разделе вы рассмотрите команды, такие как **rpm** и **yum**, в контексте серверных служб, необходимых для последующих глав. Если вы предпочитаете использовать программный инструмент **GNOME**, обратитесь к Главе 7. Как правило, вы можете использовать любой из этих вариантов для установки желаемых сервисов.

Установите сервер vsFTP с помощью команды rpm

Как правило, для установки службы требуется более одного пакета **RPM**. Единственным исключением является пакет **RPM**, связанный с сервером **vsFTP**. Для этого, если вы смонтировали DVD-диск RHEL 7 в каталоге **/media**, вы можете установить сервер **vsFTP** с помощью следующей команды (номер версии может отличаться):

```
# rpm -ivh /media/Packages/vsftpd-3.0.2-9.el7.x86_64.rpm
```

Установите серверные службы с помощью команды yum

Как обсуждалось в **Главе 7**, команда **yum** может использоваться для установки пакетов с зависимостями. Иногда зависимости просты. Например, для служб **DNS**, настроенных в главе **13**, вы можете быть лучше знакомы с **BIND**, в отличие от службы **Unbound DNS**.

Один из способов установки пакета **bind** с зависимостями - с помощью следующей команды:

```
# yum install bind
```

При необходимости вы можете использовать команду **yum install** для установки пакета способом, который автоматически идентифицирует и устанавливает все зависимые пакеты.

ТАБЛИЦА 11-3 Группы пакетов сервера, относящиеся к RHCE

Пакетная группа	Описание

Файл и сервер хранения	Группа пакетов для серверов хранения **Samba**, **NFS** и **iSCSI**.
Почтовый сервер	Пакеты поддержки для служб **SMTP** и **Internet Message Access Protocol (IMAP)**; сервисами по умолчанию являются **Postfix** и **Dovecot**. Сервер **sendmail** является необязательным пакетом в этой группе.
Сервер сетевой инфраструктуры	Группа среды для **DNS**, **rsyslog**, **Samba**, **FTP** и других сервисов; все пакеты в этой группе не являются обязательными.
Клиент сетевой файловой системы	Включает клиенты для автомонтера, **Samba** и **NFS**.
Веб сервер	Включает базовые пакеты веб-сервера **Apache**.
Сервер базы данных MariaDB	Включает только один обязательный пакет, **mariadb-server**.

Установите группы пакетов сервера с помощью команды yum

Глава 7 также описывает, как пакеты **RHEL 7** организованы в группы. У каждой из этих групп есть имена, которые можно идентифицировать с помощью команды **yum group list**. Соответствующие группы для экзамена **RHCE** перечислены в **Таблице 11-3**.

Вы можете идентифицировать различные пакеты и подгруппы в каждой группе с помощью переключателя списка групп; например, следующая команда выводит список подгрупп, входящих в группу среды **Basic Web Server**:

 # yum group info " Basic Web Server "

Ввод для **RHEL 7** показан на **рисунке 11-3.** Оттуда вы можете определить пакеты, включенные в каждую подгруппу. Например, следующая команда выводит список пакетов в группе веб-серверов:

 # yum group info web-server

РИСУНОК 11-3. Пакеты в группе среды Basic Web Server

```
[root@server1 ~]# yum group info "Basic Web Server"
Loaded plugins: langpacks, product-id

Environment Group: Basic Web Server
 Environment-Id: web-server-environment
 Description: Server for serving static and dynamic internet content.
 Mandatory Groups:
    base
    core
    web-server
 Optional Groups:
   +backup-client
   +directory-client
    guest-agents
   +hardware-monitoring
   +java-platform
   +large-systems
   +load-balancer
   +mariadb-client
   +network-file-system-client
   +performance
   +perl-web
   +php
   +postgresql-client
   +python-web
   +remote-system-management
   +web-servlet
```

Вывод показан на **рисунке 11-4**. Обратите внимание, что пакеты делятся на три категории: **обязательные, стандартные и необязательные (mandatory, default, and optional)**. Если вы выполните следующую команду, будут установлены только пакеты и группы в категориях обязательных и по умолчанию:

 # yum group install "web-server"

РИСУНОК 11-4 Пакеты в группе пакетов Веб-сервер

```
[root@server1 ~]# yum group info web-server
Loaded plugins: langpacks, product-id

Group: Web Server
 Group-Id: web-server
 Description: Allows the system to act as a web server, and run Perl and Python
web applications.
 Mandatory Packages:
    httpd
 Default Packages:
   =crypto-utils
    httpd-manual
    mod_fcgid
    mod_ssl
 Optional Packages:
   certmonger
   libmemcached
   memcached
   mod_auth_kerb
   mod_nss
   mod_revocator
   mod_security
   mod_security_crs
   perl-CGI
   perl-CGI-Session
   python-memcached
   squid
[root@server1 ~]#
```

В большинстве случаев это не проблема. Однако иногда вам может понадобиться установить пакеты, которые указаны как дополнительные. Хотя есть способы настроить установку дополнительных пакетов с помощью переключателя групповой установки, для наших целей проще просто установить необходимые пакеты по имени.

Аналогичным образом вы можете установить файловый сервер **Samba** (описано **в главе 15**) и **NFS** (описано в **главе 16**) с помощью следующей команды:

 # yum groupinstall "File and Storage Server"

В **главе 13** группа пакетов «**Сервер сетевой инфраструктуры**» включает в себя пакеты, связанные с ведением журнала и **DNS**. Однако, поскольку все пакеты в этой группе являются необязательными, команда установки **yum group** не будет устанавливать пакеты из этой группы. К счастью, пакет **rsyslog** уже установлен по умолчанию, даже при минимальной установке **RHEL 7**, но вы захотите установить **DNS** для решения одной из задач **RHCE**. Один из способов настроить **службу кэширования DNS** для главы 13 - установить Unbound DNS с помощью следующей команды:

 # yum install unbound

Для ряда серверных служб вы должны убедиться, что установлены соответствующие клиентские пакеты. Группа пакетов клиентов сетевой файловой системы может помочь в этом отношении; следующая команда установит клиенты для **автомонтера, Samba и NFS**:

 # yum group install "Network File System Client"

Другой вид сетевого сервера относится к хранилищу **iSCSI**. Интерес представляют две группы пакетов: сервер файлов и хранилищ, уже упомянутый ранее, и клиент хранилища **iSCSI**. Наконец, пара интересных пакетов не входит в стандартные группы пакетов. Они настраивают **NTP-сервер и аутентификацию** для удаленных пользовательских каталогов. Если они еще не установлены, вам необходимо установить их. Один метод с помощью следующей команды:

 # yum install ntp sssd

Мы ориентируемся на методы установки из командной строки, потому что они, как правило, самые быстрые. Конечно, вы можете установить пакеты с помощью инструмента «Установка и удаление программного обеспечения (**GUI Add/Remove Software tool**)», описанного в главе 7.

Базовая конфигурация

Хотя текущие цели **RHCE** более конкретны, чем когда-либо, лучше сохранить то, что вы меняете, как можно проще. Как указано в целях, вас попросят «настроить службу для основной операции». Базовую операцию проще настроить. Это часто более безопасно. Если вы делаете меньше для настройки службы, это займет меньше времени. У вас будет больше шансов закончить экзамен. Вы сможете сделать больше на работе.

Детали, связанные с базовой конфигурацией, описаны в следующих главах.

Убедитесь, что установленная служба переживает перезагрузку

В главе 5 вы рассмотрели, когда служба запускается или не запускается во время процесса загрузки. Самый простой метод связан с командой **systemctl**. Для обзора **systemctl list-unit-files --type=service** команда выводит список всех сервисных модулей и их активацию при загрузке. Для служб, описанных в следующих главах, после установки соответствующих пакетов вы должны убедиться, что они запускаются во время процесса загрузки с помощью следующих команд:

 # systemctl enable httpd
 # systemctl enable iscsi
 # systemctl включить mariadb
 # systemctl enable nfs-server
 # systemctl enable nmb
 # systemctl enable ntpd
 # systemctl enable rsyslog
 # systemctl enable smb
 # systemctl enable sshd
 # systemctl enable target
 # systemctl enable unbound

Это просто список. На реальном экзамене установите только те службы, которые вам предлагается установить.

Конечно, во время экзамена вам может быть предложено убедиться, что служба не запускается во время процесса загрузки. Кроме того, имейте в виду, что в производственной среде установка такого количества сервисов в одной системе происходит редко из-за угроз безопасности.

Обзор доступа через уровни безопасности

Первое место, чтобы проверить сервис из локальной системы. Например, если вы можете подключиться к серверу **Apache** из этой системы, вы настроили базовую конфигурацию **Apache**.

Если у вас есть проблемы с локальным или удаленным подключением, у вас могут быть проблемы, связанные с **SELinux** или **различными брандмауэрами** на основе пользователей и хостов. По вопросам, выходящим за пределы **SELinux**, обратитесь к сетевым инструментам команд, установленным в **главе 2: telnet, elinks и nmap**.

Устранение неполадок SELinux

Если конфигурация настроена правильно, но все еще не работает, возможно проблема в **SELinux**, как правило, в одной из двух следующих областей:

- **Логические настройки(Boolean settings) -** Например, чтобы разрешить серверу **Apache** доступ к домашним каталогам пользователей, включите **логические переменные(флаги)** для **SELinux httpd_enable_homedirs**.
- **Контексты файлов SELinux -** Убедитесь, что контексты файлов и каталогов совпадают с контекстами каталогов по умолчанию. Предположим, вы настроили виртуальный веб-хост в каталоге **/virtual/host**. Запустите команду **ls -Z /virtual/host**. Контексты файлов, которые вы видите в этих выходных данных, должны совпадать с тем, что вы видите из команды ls -Z **/var/www/html**.

Затем проверьте соединение с удаленной системой:

Устранение неполадок (Zone-Based Firewall) зонного межсетевого экрана

Если система разрешает доступ для связи с сервером в зону по умолчанию, вы увидите это в выводе команды **firewall-cmd --list-all**. Чтобы просмотреть конфигурацию для всех зон, запустите **firewall-cmd --list-all-zone**.

Хотя вы можете использовать инструмент настройки брандмауэра, описанный **в главах 4 и 10**, вам нужно знать, как настроить брандмауэры из командной строки.

Если порт или сервер не открыт в брандмауэре, попытка подключения к службе отклоняется. Например, для сервера **SSH** вы можете получить следующее сообщение:

ssh: connect to host server1.example.com port 22: No route to host

Чтобы проверить работоспособность подключения к удаленной службе, вы можете использовать команду **telnet** или **nmap**. Например, выполните следующую команду, чтобы проверить подключение к порту **HTTP** на **сервере 192.168.122.50**:

$ telnet 192.168.122.50 80

Если вы можете успешно подключиться к серверу, вы увидите следующий ответ:

Escape character is '^]'

Точно так же вы можете использовать **nmap**, как показано ниже, для проверки подключения к службе **HTTP** через **порт TCP 80**:

$ nmap -p 80 192.168.122.50

Если вы можете успешно подключиться к услуге, вы увидите следующий вывод:

PORT STATE SERVICE
80/tcp open http

УПРАЖНЕНИЕ 11-3

Практика устранения неполадок сетевых подключений

В этом упражнении мы исследуем влияние различных неправильных настроек сети и **firewalld** на работающий сервис. Мы предполагаем, что у вас есть работающая служба **SSH**, работающая на **server1.example.com.**

1. С другого хоста выполните команду **ping 192.168.122.50**, чтобы проверить соединение с сервером.
2. Теперь выполните следующую команду на сервере **server1**:

   # systemctl stop network

   Запустите команду ping еще раз. Что вы видите в выводе результата выполнения команды?
3. Восстановите сетевое подключение с помощью **systemctl start network**.
4. С компьютера клиента используйте команду **telnet** или **nmap** для проверки соединения на порту сервера **SSH**:

   $ telnet 192.168.122.50 22

   В случае успеха вы увидите следующий вывод:

   Escape character is '^]'

   Введите команду **quit**. Вы должны увидеть сообщение об ошибке от сервера **OpenSSH**, за которым следует это сообщение:

   Connection closed by foreign host

   Заблокируйте подключение к службе **SSH** на сервере **server1** с помощью следующей команды:

   # firewall-cmd --remove-service = ssh

5. Попробуйте снова выполнить команды **ping** и **telnet**. Какой выход вы видите?
6. Восстановите соединение на брандмауэре, запустив **firewall-cmd --reload.**
7. Заблокируйте **IP-адрес клиента (предполагая, что это 192.168.122.1),** как показано здесь:

   # firewall-cmd --add-rich-rule='rule family=ipv4 source address=192.168.122.1 drop'

8. Попробуйте снова выполнить команды **ping** и **telnet**. Какой вывод вы видите?

В общем случае, если команда **telnet** или **nmap** не подключается к указанному порту, у вас может быть одна из следующих проблем брандмауэра:

- Брандмауэр на основе зоны межсетевого экрана (**firewalld zone-based**) может блокировать нужный порт.

- Брандмауэр на основе зоны межсетевого экрана (**firewalld zone-based**) может ограничивать доступ к клиенту.
- Система **TCP Wrappers**, обсуждаемая в этой главе, также может ограничивать доступ к конкретным клиентам и пользователям по сервисам.
- Некоторые серверы **содержат файлы** конфигурации, которые также ограничивают доступ на основе **пользователей, IP-адресов и имен хостов**.

Устранение неполадок брандмауэра TCP Wrappers

Напротив, если служба защищена **TCP Wrappers**, поведение сообщения об ошибке будет другим. Для этого раздела мы настроили файлы **/etc/hosts.allow** и **/etc/hosts.deny** в системе **server1.example.com**, чтобы разрешить доступ только из систем **.example.com** в сети **192.168.122.0/24**. Это означает, что доступ с систем, таких как **outsider1.example.org**, по **IP-адресу 192.168.100.100** запрещен.

В этом случае, когда мы попытались получить доступ к системе **server1.example.com** с помощью команды **ssh**, мы получили следующее сообщение об ошибке:

ssh_exchange_identification: Connection closed by remote host

Напротив, команда **telnet server1.example.com 22** из той же системы возвращает следующие сообщения, которые на мгновение останавливаются:

Trying 192.168.122.50
Connected to server1.example.com.
Escape character is '^]'

В течение нескольких минут кажется, что система собирается подключиться, но затем блок из TCP Wrappers приводит к следующему сообщению:

Connection closed by foreign host.

УПРАЖНЕНИЕ 11-4

Обзор различных эффектов firewalld и TCP Wrappers

В этом упражнении предполагается использование рабочего сервера **vsFTP**, аналогичного тому, который был настроен для установки в **главе 1**. Настройте этот сервер **vsFTP** в системе **server1.example.com**. Убедитесь, что брандмауэр блокирует трафик на стандартном **порту FTP, TCP 21**, а затем проверьте подключение из заблокированной системы **outsider1.example.org**. Для обзора эти системы, настроенные в **главах 1 и 2**, имеют **IP-адреса 192.168.122.50** и **192.168.100.100** соответственно.

Затем откройте **TCP-порт 21** на брандмауэре. Кроме того, ограничьте доступ с помощью **TCP Wrappers**.

Это упражнение сложное; каждый пронумерованный шаг требует нескольких команд или действий. В некоторых случаях требуется указанная команда.

1. Если он еще не установлен, установите сервер **vsFTP**, как описано в главе. Убедитесь, что сервер активен с помощью команды **systemctl start vsftpd**.
2. Запустите средство настройки брандмауэра с помощью команды **firewall-config**. Убедитесь, что **FTP** не активирован в списке служб в зоне по умолчанию. Убедитесь, что изменения вступили в силу, а затем выйдите из средства настройки брандмауэра.
3. Попробуйте подключиться к серверу **vsFTP** из локальной системы с помощью команды, такой как **lftp localhost**. Он должен работать, что вы можете подтвердить из командной

строки **lftp localhost:/>** с помощью команды **ls**. Выйдите из сервера **vsFTP** с помощью команды **quit**.

4. Перейти к системе **outsider1.example.org**. Допустимо подключаться к нему через **SSH**; на самом деле, это может быть единственный доступный метод подключения к этой системе на экзамене (и в реальной жизни).
5. Попробуйте проверить связь с системой, на которой запущен сервер **vsFTP**, с помощью команды **ping 192.168.122.50**. Не забудьте нажать **Ctrl-C**, чтобы остановить процесс. Попробуйте подключиться к серверу **vsFTP** с помощью **команды lftp 192.168.122.50**. Что происходит? Попробуйте подключиться к системе с помощью команды **telnet 192.168.122.50 21**. Что просходит?
6. Вернитесь в систему **server1.example.com**. Снова откройте инструмент настройки брандмауэра и на этот раз сделайте **FTP** доверенной службой. Не забудьте применить изменения перед выходом из инструмента настройки брандмауэра.
7. Откройте файл **/etc/hosts.allow** и включите следующую запись:

   vsftpd: localhost 127. 192.168.122.50

8. Откройте файл **/etc/hosts.deny** и включите следующую запись:

   vsftpd: ALL

9. Вернитесь в систему **outsider1.example.com**, как описано в шаге 4. Повторите шаг 5. Что происходит после каждой попытки подключения?
10. Вернитесь в систему **server1.example.com**. Откройте **/etc/hosts.allow** и **/etc/hosts.deny** и удалите строки, созданные в шагах 7 и 8.
11. Еще раз, перейдите в систему **outsider1.example.org**. Повторите шаг 5. Обе команды должны привести к успешному соединению. Команда **quit** должна завершиться в обоих случаях.
12. БОНУС: Просмотрите соединения через содержимое файла **/var/log/secure**. Просмотрите **исходные IP-адреса** в этом файле. Используйте эту информацию для настройки **firewalld**, чтобы запретить доступ ко всем, кроме **одного IP-адреса**.

СЦЕНАРИЙ И РЕШЕНИЕ	
Вы хотите ограничить **доступ по SSH** для двух пользователей.	Укажите нужные имена пользователей в файле конфигурации сервера **SSH**, **/etc/ssh/sshd_config**, с помощью директивы **AllowUsers**.
Вам нужно ограничить **доступ по SSH системам в сеть 192.168.122.0/24**.	Вы можете использовать **TCP Wrappers**. Сконфигурируйте **/etc/hosts.allow**, чтобы разрешить доступ к демону **sshd** из систем в указанной сети. Сконфигурируйте **/etc/hosts.deny**, чтобы ограничить доступ к **sshd** из **ALL** систем.
Вы должны убедиться, что пользователь и типы файлов **SELinux** выдержат в **relbel(перемаркировку)**.	Используйте команду **semanage fcontext -a**, чтобы указать нужного пользователя и типы файлов для нужных каталогов.
Вам необходимо запустить **Apache** на нестандартном сетевом порту.	Измените определение порта с помощью порта **seemanag -a**. Не забудьте настроить службу для работы на другом порту и проверить правила брандмауэра.
Сервер доступен только локально.	Проверьте параметры безопасности для правил **firewalld** и **TCP Wrappers**;

	Убедитесь, что служба разрешает удаленный доступ.
Сервер правильно настроен, но по-прежнему недоступен.	Проверьте логические значения **SELinux booleans** и **типы меток (boolens)**файлов.

РЕЗЮМЕ СЕРТИФИКАЦИИ

В этой главе основное внимание уделено общим шагам, необходимым для настройки, защиты и доступа к различным службам. Демоны управляются файлами модулей в каталоге **/lib/systemd/system** и файлами конфигурации в **/etc/sysconfig**. Доступ к различным аспектам серверных служб может контролироваться разными логическими элементами **SELinux**.

Файлы конфигурации **сервера SSH** находятся в каталоге **/etc/ssh**. Файл конфигурации **sshd_config** содержит значительное количество параметров для настройки этой службы.

Чтобы настроить службу, вам нужно установить нужные пакеты и убедиться, что служба активна после следующей перезагрузки. Вам также нужно будет перемещаться по различным доступным параметрам безопасности, включая **SELinux**, межсетевые экраны на основе зон и безопасность на основе **TCP Wrappers** в файлах **/etc/hosts.allow** и **/etc/hosts.deny.**

Двухминутная тренировка

Ниже приведены некоторые ключевые моменты целей сертификации в главе 11.

Конфигурация системы Red Hat

- Системные службы могут быть запущены **systemctl** на основе файлов конфигурации модуля в
- Каталогах **/lib/systemd/system** и **/etc/systemd/system.**
- Системные службы используют файлы базовой конфигурации в каталоге **/etc/sysconfig**. Такие файлы часто включают основные параметры для сервисных демонов.
- При настройке сетевого сервера вы должны быть обеспокоены **логическими значениями (boolens) SELinux**, **межсетевые экраны на основе зон (zone-based firewalls)**, **TCP Wrappers** и многое другое.
- Доступность сервисов на серверах должны тестироваться локально и удаленно.

Linux с улучшенной безопасностью SELinux

- Отдельные сервисы часто защищены несколькими логическими (**booleans**) значениями **SELinux**.
- Логические значения **SELinux** хранятся в каталоге **/sys/fs/selinux/booleans** с описательными именами файлов.
- Логические значения **SELinux** можно изменить с помощью команды **setsebool -P** или **Инструмент управления SELinux**. В командной строке обязательно **используйте ключ -P**; в противном случае изменения не переживут перезагрузку.
- **Контексты файла SELinux** можно изменить с помощью команды **chcon**. Тем не менее изменение не переживает пере маркировку (**relabel)**, если новое правило контекста не сделано постоянным командой **semanage fcontext -a**. Изменения документированы в **file_contexts.local** в каталоге **/etc/selinux/target/contexts/files.**
- **Метки порта SELinux** можно изменить с помощью команды **semanage port -a**, чтобы сервисы могли слушать нестандартные сетевые порты.

Пакет SSH

- Файлы конфигурации сервера **SSH** в каталоге **/etc/ssh** включают файлы клиента и сервера, вместе с открытыми и частными парами ключей хоста **RSA и ECDSA.**
- Файл конфигурации сервера **SSH, sshd_config**, может быть настроен на основе пользователя безопасность.
- Директива **AllowUsers** в **sshd_config** указывает, каким пользователям разрешено входить в систему через **SSH**.
- Самый простой способ настроить **безопасность SSH на основе хоста - через TCP Wrappers**.

Контрольный список безопасности и конфигурации

- Для подготовки к экзамену **RHCE** вам необходимо установить ряд служб такие команды, как **rpm** и **yum**.
- Один из способов убедиться, что сервисы выживают после перезагрузки, - это команда **systemctl**; полный список таких команд, относящихся к услугам **RHCE**, приведен в главе.
- Вам необходимо настроить доступ к службе через уровни безопасности, включая **SELinux, зональные межсетевые экраны и TCP Wrappers.**

САМОПРОВЕРКА

Следующие вопросы помогут оценить ваше понимание материалов, представленных в этой главе. Поскольку на экзаменах **Red Hat** не появляется вопросов с несколькими вариантами ответов, вопросы с несколькими вариантами ответов не появляются в этой книге. Эти вопросы исключительно проверяют ваше понимание главы. Это нормально, если у вас есть другой способ выполнения задачи. Получение результатов, а не запоминание пустяков - вот на что рассчитывает **Red Hat** Экзамены. На многие из этих вопросов может быть более одного ответа.

Конфигурация системы Red Hat

1. В какой каталог входят файлы конфигурации, которые определяют параметры запуска демонов для различных служб?

_____

2. Какая команда перезагружает конфигурацию сервера **SSH** без остановки службы?

_____

Linux с улучшенной безопасностью (SELinux)

3. Какой каталог содержит логические параметры, связанные с **SELinux**? Укажите полный путь.

_____

4. Какая страница руководства содержит параметры **SELinux**, связанные с демонами **NFS**?

_____

5. Какая команда восстанавливает контекст файла **SELinux** по умолчанию для данного каталога?

_____

6. Какой файл изменяется при запуске команды **semanage fcontext -a**? Подсказка: это в Каталоге **/etc/selinux/target/contexts/files**.

_____

7. Какая команда перечисляет текущую конфигурацию метки порта **SELinux для сервиса MariaDB (MySQL)?**

_____

Secure Shell Server

8. Какой каталог содержит файл конфигурации сервера **OpenSSH** и ключи хоста?

_____

9. Какая директива указывает номер порта локального **SSH-сервера** в связанном конфигурационный файл?

_____

10. Какая директива указывает список разрешенных пользователей в файле конфигурации сервера **SSH**?

_____

Контрольный список безопасности и конфигурации

11. Какая команда отображает список доступных групп среды?

_____

12. Какая команда может помочь службе abcd пережить перезагрузку?

_____

LAB ВОПРОСЫ

Некоторые из этих лабораторий включают в себя упражнения по настройке. Вы должны делать эти упражнения на тестовых машинах только. Предполагается, что вы выполняете эти упражнения на виртуальных машинах на основе KVM.

Red Hat представляет свои экзамены в электронном виде. По этой причине лаборатории в этой и будущих главах доступно из носителей, сопровождающих книгу, в подкаталоге Chapter11 /. Если вы еще не настроили **RHEL 7 в системе, обратитесь к главе 1 за инструкциями по установке**.

Ответы для лабораторных работ следуют за ответами самопроверки для вопросов, которые нужно заполнить.

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 11. Большинство лабораторных работ для этой главы просты и требуют очень мало команд или изменений в одном или двух файлах конфигурации.

Лабораторная работа 1

Целью этой лабораторной работы является демонстрация некоторых возможностей файлов в каталоге **/etc/sysconfig**. Это возможность для вас поэкспериментировать, посмотреть, как такие файлы можно использовать для изменения способа запуска службы.

В этой лабораторной работе вы будете работать с файлом **/etc/sysconfig**, связанным с **веб-сервером Apache, httpd**. Как и в случае любого другого файла конфигурации, вы должны сделать его резервную копию перед внесением каких-либо изменений. Если вы еще не установили веб-**сервер Apache**, следуйте соответствующим инструкциям в теле главы 1.

1. Сделайте резервную копию текущей версии **/etc/sysconfig/httpd**.
2. Убедитесь, что **веб-сервер Apache** работает; Вы можете перезапустить его с помощью команды **systemctl restart httpd**. Просмотрите текущие **процессы Apache** с помощью **команды ps aux | grep httpd**. Сколько процессов **Apache** в настоящее время работает?
3. Откройте файл **/etc/sysconfig/httpd** и добавьте следующую директиву:

   OPTIONS="-l"

4. Просмотрите страницу руководства **httpd**. Как вы думаете, этот вариант будет делать?
5. Перезапустите **веб-сервер Apache**. Что просходит?
6. Есть ли в данный момент какие-то **процессы Apache**?
7. Восстановите исходную версию файла **/etc/sysconfig/httpd**.
8. Будь смелым. Предполагая, что вы сохранили резервную копию файла конфигурации **/etc/sysconfig/httpd**, попробуйте использовать другую опцию, основанную на шагах с 4 по 8.

Лабораторная работа 2

В этой лабораторной работе вы настроите аутентификацию на основе **открытого/закрытого ключа** между двумя системами **RHEL 7**. Любые две системы будут работать для этой цели. Просто будьте готовы использовать те же две системы в лабораториях 3, 4 и 5. При подготовке к этой лаборатории убедитесь, что у каждой системы есть обычный пользователь с именем **hawaii**. Используйте следующую кодовую фразу:
I love Linux!

Лабораторная работа 3

Повторите лабораторную работу 2, но с двумя отличиями: настройте пару ключей **ECDSA** и настройте пользователя по имени **tonga** на клиенте, настроенного для подключения к учетной записи пользователя **hawaii** на сервере.

Лабораторная работа 4

В этой лабораторной работе вы настроите безопасность на уровне пользователя на сервере **SSH**. Используйте те же системы, которые были настроены для **лабораторий 2 и 3**. Ограничьте доступ для пользователя **hawaii**, настроенного в **лаборатории 2**, а затем попытайтесь получить доступ к другой учетной записи на **сервере SSH**. Попробуйте снова получить доступ к учетной записи администратора **root на сервере SSH**.

Лабораторная работа 5

В этой лабораторной работе вы настроите **SSH-сервер**, настроенный в предыдущих лабораторных работах, на **TCP-порт 8122**. После этого перезагрузите службу. Попробуйте подключиться локально. Настройте системы так, чтобы вы также могли подключаться из удаленной системы. В качестве дополнительной задачи повторите то же упражнение, используя **порт 8022**. Что происходит?

Лабораторная работа 6

В этой лабораторной работе вы настроите каталог **/virtual/web** с тем же контекстом файла **SELinux**, что и каталог **/var/www**. Кроме того, настройте каталог **/virtual/web/cgi-bin** с тем же контекстом файла **SELinux**, что и каталог **/var/www/cgi-bin**.

ОТВЕТЫ НА САМОПРОВЕРКУ

Конфигурация системы Red Hat

1. Небольшой вопрос с подвохом: файл в каталоге **/etc/sysconfig**, а также файлы модулей в **/lib/systemd/system** и **/etc/systemd/system**, могут указывать опции для различных сервисных демонов при запуске.
2. Команда для перезагрузки конфигурации службы SSH:

   # systemctl reload sshd

   Linux с улучшенной безопасностью

3. Каталог с булевыми значениями **SELinux** - это **/sys/fs/selinux/booleans**.
4. Страница **man nfsd_selinux** содержит некоторые логические значения **SELinux** для этой службы.
5. Команда, которая восстанавливает контекст файла по умолчанию в данном каталоге, является **restorecon**.
6. Имя файла, который изменяется указанной командой, - **file_contexts.local**.
7. Один приемлемый ответ

   #semanage port -l | grep mysql

   Secure Shell Server

8. Файл конфигурации сервера **OpenSSH** и ключи хоста находятся в каталоге **/etc/ssh**.
9. Директива **Port**.
10. Директива **AllowUsers**.

    Контрольный список безопасности и конфигурации
11. Команда, в которой перечислены все доступные группы среды, это **yum group list**
12. Предполагая, что служба **abcd** также связана с единицей службы в **/lib/systemd/system** каталоге, команда, которая помогла бы ему пережить перезагрузку, является **systemctl enable abcd**.

ОТВЕТЫ ЛАБАРОТОРНОЙ РАБОТЫ

Лаборатория 1

Эта лабораторная работа должна дать вам представление о том, что можно сделать с файлами **/etc/sysconfig** и как эти файлы способны изменить способ запуска демона. Эта лаборатория должна также продемонстрировать риски; неправильное изменение, такое как показано в лаборатории, означает, что служба не будет работать.

Лаборатория 2

Хотя аутентификация на основе ключей **SSH** была рассмотрена в первой части этой книги, она также является требование к экзамену **RHCE**. Если вы не помните, как настроить аутентификацию на основе ключей, просмотрите **главу 4**. В этой лаборатории есть три показателя успеха:

- В каталоге клиента **/home/hawaii/.ssh** будет файл **id_rsa** и файл **id_rsa.pub**.
- Вы сможете подключиться к удаленной системе без пароля. Просто введите «**I love Linux!**» пароль (без кавычек) при появлении запроса.
- Вы найдете содержимое файла **id_rsa.pub** пользователя в удаленном файле **authorized_keys** в **/home/hawaii/.ssh** каталоге.

Небезопасные разрешения являются одной из наиболее распространенных причин сбоя аутентификации на основе ключей SSH. Ваш каталог **~ /.ssh** должен иметь восьмеричные разрешения **0700**, тогда как **закрытый ключ** и **uthorized_keys** для файлов ключей должны быть установлены биты прав доступа **0600**.

Лаборатория 3

Как и в **лаборатории 2**, в этой лаборатории есть три показателя успеха:

- В каталоге клиента **/home/tonga/.ssh** будет файл **id_ecdsa** и файл **id_ecdsa.pub**.
- Вы сможете подключиться к удаленной системе без пароля. Просто введите ключевую фразу «**I love Linux!**» при появлении запроса.
- Вы найдете содержимое клиентского файла **id_ecdsa.pub** в удаленном файле **author_keys** в **/home/hawaii/.ssh** каталог.

Лаборатория 4

Самый простой способ реализовать эту лабораторную работу - добавить следующую директиву в файл **/etc/ssh/sshd_config:**

AllowUsers hawaii

Только не забудьте перезагрузить или перезапустить службу **SSH** после внесения изменений; в противном случае другие пользователи все еще будут иметь доступ.

Если вам интересно, пользователь **tonga** на клиенте по-прежнему может получить доступ к учетной записи **hawaii** в **SSH** сервер с парольной фразой, поскольку разрешены подключения к учетной записи **hawaii** пользователя. Личность удаленной учетной записи не имеет значения для директивы **AllowUsers**.

Если вы внесли слишком много изменений в файл **/etc/ssh/sshd_config** и хотите начать все сначала, переместите этот файл и выполните команду **yum reinstall openssh-server**. Он установит свежую копию конфигурационного файла. Если вы хотите подключиться с других учетных записей в будущем, убедитесь, что директива **AllowUsers hawaii** выключена.

О да, вам нужно было активировать директиву **PermitRootLogin no**, чтобы предотвратить вход SSH в учетная запись **root**?

Лаборатория 5

Успех в этой лабораторной работе подтверждается хорошим **SSH**-соединением между клиентом и сервером. Если вы просто хотите убедиться, выполните команду **ssh -p 8122** от клиента. Если вы не отключили директиву **AllowUsers** на сервере, это соединение должно быть к учетной записи **hawaii**.

Кроме того, эта лаборатория должна дать вам представление об усилиях, необходимых для настройки малоизвестных портов. Тем немение, хотя команда **nmap** обнаружит прослушивающее приложение на порту 8122, сервис использующий порт будет неясным; соответствующий результат будет

PORT STATE SERVICE
8122/tcp open unknown

Перейдите в систему клиента и попробуйте подключиться к серверу **SSH**. Помните, вам также нужно открыть **порт 8122 в брандмауэре сервера SSH**.

Хотя повторение этой лабораторной работы с портом **8022** может выглядеть аналогично использованию **порта 8122**, существует небольшая проблема
при попытке добавить **порт 8022 к метке ssh_port_t**:

#semanage port -a -t ssh_port_t -p tcp 8022
ValueError: Port tcp/8022 already defined

Эта ошибка происходит, потому что порт **8022** уже используется другой службой:

#semanage port -l | grep 8022
oa_system_port_t tcp 8022
oa_system_port_t udp 8022

Нет простого способа добавить порт **8022** к типу **ssh_port_t** без перекомпиляции политики. После завершения этой лабораторной работы восстановите исходные номера портов на клиенте и сервере **SSH**.

Лаборатория 6

Подтверждение успеха в этой лаборатории просто. Запустите команды **ls -Zd** для указанного каталога. **Контексты SELinux** для каталогов **/virtual/web** и **/var/www** должны совпадать со следующими контекстами:

system_u: object_r: httpd_sys_content_t: s0

Контексты для каталогов **/virtual/web/cgi-bin** и **/var/www/cgi-bin** также должны совпадать:

system_u: object_r: httpd_sys_script_exec_t: s0

Само собой разумеется, что любые изменения, которые вы делаете, должны пережить пере маркировку (**relabel**) **SELinux**. Иначе, как вы ожидаете получить кредит на свою работу? Если вы запустили команду **semanage fcontext -a** в правильных каталогах вы увидите эти контексты, перечисленные в файле **file_contexts.local**, в каталоге **/etc/selinux/target/contexts/files**:

/virtual/web(/.*)? system_u: object_r: httpd_sys_content_t: s0
/virtual/web/cgi-bin(/.*)? system_u: object_r: httpd_sys_script_exec_t: s0