Глава 10

Учебник по безопасности

ЦЕЛИ СЕРТИФИКАЦИИ

10.01. Уровни безопасности Linux
10.02 Брандмауэры и трансляция сетевых адресов
10.03 TCP Wrappers
10.04 Сменные модули аутентификации
10.05 Безопасные файлы и многое другое с GPG2
✓ Двухминутная тренировка
Q & A Самопроверка

Когда вы начнете первую главу раздела RHCE этой книги, вы начнете с безопасности. Многие администраторы и предприятия переходят на Linux, потому что считают его более безопасным. Поскольку большая часть программного обеспечения Linux выпускается по лицензиям с открытым исходным кодом, исходный код доступен для всех. Некоторые считают, что это дает преимущества хакерам, которые хотят взломать систему.
Тем не менее, разработчики Linux верят в сотрудничество. «Закон Линуса», по словам светочника с открытым исходным кодом Эрика Рэймонда, заключается в том, что «при достаточном количестве глазных яблок все ошибки мелкие». Некоторые из этих глаз принадлежат Агентству национальной безопасности США (АНБ), которое внесло много кода. в Linux, в том числе основы SELinux.

АНБ также внесло ряд других концепций, адаптированных Red Hat, которые были интегрированы в многоуровневую стратегию безопасности. К ним относятся рекомендации по настройке брандмауэров, упаковщиков пакетов и безопасности по услугам. Они охватывают как безопасность пользователей, так и хостов. Они включают в себя средства управления доступом, такие как владение, разрешения и SELinux. (Ряд этих уровней был рассмотрен в предыдущих главах.) Здесь также рассматриваются основы этих уровней безопасности, поскольку они применяются к целям RHCE.

В этой главе вы познакомитесь с некоторыми инструментами, предоставляемыми RHEL для управления безопасностью. Вы начнете с некоторых основ и продолжите детальный анализ межсетевых экранов на основе зон, сменных модулей аутентификации (PAM), TCP Wrappers и многого другого.

Это не единственная глава, посвященная безопасности. Строго говоря, он охватывает только две цели RHCE. Однако в этой главе рассматриваются темы, связанные с безопасностью в системах Linux, и эти темы могут помочь вам понять параметры безопасности, связанные с каждым сервисом в этой книге.

ЦЕЛЬ СЕРТИФИКАЦИИ 10.01

Слои безопасности Linux

Лучшая компьютерная безопасность поставляется в слоях. Если в одном слое есть нарушение, такое как проникновение через брандмауэр, скомпрометированная учетная запись пользователя или переполнение буфера, которое портит службу, почти всегда существуют другие меры безопасности, которые предотвращают или, по крайней мере, сводят к минимуму дальнейший ущерб.

ВНУТРИ экзамена

Внутри экзамена

Эта глава является первой в этой книге, посвященной требованиям **RHCE**. Как описано в целях **RHCE**, безопасность начинается с фильтрации пакетов и NAT, разработанных с помощью межсетевого экрана на основе зоны межсетевого экрана. Соответствующая цель:

- Использование **firewalld** и связанных с ним механизмов, таких как расширенные правила, зоны и пользовательские правила, для реализации фильтрации пакетов и настройки преобразования сетевых адресов (**NAT**).

Но, как предлагается во введении, безопасность является проблемой для всех охватываемых служб. в целях **RHCE**. Эта глава служит основой для обсуждения вопросов безопасности, в том числе нескольких способов

- Настройки безопасности на основе хоста и пользователя для сервиса.

При этом безопасность на основе хоста может начинаться с зонных брандмауэров, безопасности хоста и пользователя меры могут включать **TCP Wrappers** и сменные модули аутентификации.

Эти параметры начинаются с минимально настроенных бастионных хостов, которые минимизируют функциональность, связанную с отдельной системой **Linux**. Помимо брандмауэра и **SELinux** существуют опции безопасности, связанные с отдельными сервисами. Параметры изоляции, такие как изоляция **chroot**, обычно настраиваются как часть службы. Ряд этих вариантов основан на рекомендациях АНБ.

Хотя разделы о бастионных системах предназначены для ознакомления с мерами безопасности, используемыми для служб уровня RHCE, они также включают в себя те параметры безопасности, которые часто связаны с экзаменом RHCSA, которые описаны в предыдущих главах.

Бастионные системы

Правильно настроенная бастионная система сводит к минимуму риск нарушения безопасности. Он основан на минимальной установке с меньшим количеством программного обеспечения, чем было установлено в системах, настроенных в главах 1 и 2. Бастионная система настроена с двумя службами. Один сервис определяет функциональность системы. Это может быть веб-сервер, файловый сервер, сервер аутентификации или что-то подобное. Другая служба поддерживает удаленный доступ, такой как SSH или, возможно, VNC через SSH.

До виртуализации использование бастионных систем часто было ограничено. Только самые богатые предприятия могли позволить себе выделять разные физические системы для каждой услуги. Если требовалась избыточность, затраты только увеличивались.

Благодаря виртуализации бастионные системы доступны даже для небольших предприятий. Все, что нужно, это стандартная минимальная установка. С помощью нескольких файлов кикстарта вы, как администратор такой сети, можете легко создать целую группу бастионных систем. Затем каждая система может быть настроена и выделена для одного сервера.

Хорошо построенные бастионные системы следуют двум принципам:

- Если вам не нужно программное обеспечение, удалите его.
- Если вам нужно программное обеспечение, но вы его не используете, убедитесь, что оно не активно.

В общем, хакеры черной шляпы не могут воспользоваться недостатком безопасности, если соответствующая служба не установлена. Если вам необходимо установить службу в целях

тестирования, оставьте эту службу неактивной. Это может помочь свести риски к минимуму. Разумеется, брандмауэры, настроенные для каждой бастионной системы, должны пропускать трафик только для выделенной службы и метода удаленного доступа.

Лучшие защиты с обновлениями безопасности

Обновления безопасности чрезвычайно важны. Вы можете просмотреть доступные обновления с помощью инструмента «Обновление программ». Вы можете запустить этот инструмент в графическом интерфейсе с помощью команды **gpk-update-viewer**. Как уже говорилось в главе 7, вы можете настроить автоматические обновления безопасности с помощью инструмента «Настройки обновлений программного обеспечения», который можно запустить в графическом интерфейсе с помощью команды **gpk-prefs**.

На практике безопасность часто является гонкой между обнаружением уязвимости и появлением обновления. Пока эти обновления не установлены, любые уязвимые службы могут быть уязвимы.

Как профессионал **Linux**, вы должны знать об этих уязвимостях. Если вы поддерживаете такие серверы, как **Apache, vsFTP и Samba**, следите за потоками информации от этих разработчиков. Новости безопасности могут поступать в различных формах, от обновлений доски объявлений до RSS-каналов. Обычно Red Hat также идет в ногу со временем по таким вопросам. Однако, если вы подписались на форумы, поддерживаемые разработчиками службы, лучше всего узнавать о проблемах и планируемых решениях непосредственно из источника. В какой-то степени это область безопасности, ориентированной на конкретные услуги.

Уязвимости информационной безопасности отслеживаются в стандартизированной системе форматов, известной как **Common Vulnerabilities and Exposures (CVE),** которая поддерживается корпорацией **MITER (http://cve.mitre.org).** В своих рекомендациях по безопасности «Ошибки» Red Hat всегда ссылается на соответствующие идентификаторы CVE. Вам следует ознакомиться с форматом CVE и следить за обновлениями базы данных Red Hat CVE и веб-сайтов объявлений Errata, доступных по **адресу https://access.redhat.com/security/cve и https://rhn.redhat.com/errata**, соответственно.

Сервис-специфическая безопасность

Большинство основных служб имеют некоторый уровень безопасности, который можно настроить внутри. Во многих случаях вы можете настроить службу для ограничения доступа по хосту, сети, пользователю и группе. Как указано в целях **RHCE**, вам необходимо знать, как настроить безопасность на уровне хоста и пользователя для каждой перечисленной службы. Также доступны параметры **SELinux**, которые могут помочь защитить каждую из этих служб. Хотя подробности обсуждаются в соответствующих последующих главах, ниже приведен краткий обзор параметров безопасности для конкретной службы.

HTTP/HTTPS Сервис-специфическая безопасность

Хотя есть альтернативы, основной услугой для протоколов **HTTP и HTTPS в Linux** является веб-сервер **Apache**. Фактически, Apache является доминирующим веб-сервером в Интернете. Несомненно, файлы конфигурации **Apache** являются сложными, но они должны быть, потому что проблемы безопасности в Интернете являются существенными. Некоторые варианты ответа на эти проблемы рассматриваются в главе 14.

Apache включает в себя множество дополнительных программных компонентов. Не устанавливайте больше, чем это абсолютно необходимо. Если в скрипте **Common Gateway Interface (CGI)** есть нарушение безопасности, и вы не установили поддержку **Apache** для скриптов **CGI**, эта проблема безопасности вас не затронет. Однако, поскольку **RHCE** определяет цель развертывания «базового приложения CGI», у вас нет такой роскоши для сдачи экзамена.

К счастью, с **Apache** вы можете ограничить доступ несколькими способами. Лимиты могут быть созданы на сервере или на отдельных виртуальных хостах. Различные ограничения могут быть созданы на обычных и безопасных веб-сайтах. Кроме того, **Apache** поддерживает **использование безопасных сертификатов**.

DNS-сервис-специфическая безопасность

Серверы службы доменных имен (**DNS**) являются большой целью для хакеров черной шляпы. Имея это в виду, **RHEL** 7 включает в себя пакет **bind-chroot**, который настраивает необходимые файлы, устройства и библиотеки в изолированном подкаталоге. Этот подкаталог предоставляет ограничение для любого пользователя, который нарушает безопасность **DNS**, известную как тюрьма **chroot**. Он предназначен для ограничения каталогов, в которых хакер черной шляпы может перемещаться, если он действительно взломает службу. Другими словами, если кто-то взломает **DNS-сервер RHEL 7**, он не сможет «убежать» из подкаталога, настроенного **как chroot-тюрьма**.

Поскольку от кандидатов на экзамен **RHCE** не ожидается создание главного или подчиненного **DNS-сервера**, проблемы и риски несколько ограничены. Тем не менее, в главе 13 вы узнаете, как ограничить доступ к настроенному **DNS-серверу хостом**.

NFS-сервисная безопасность

С переходом на версию 4 сетевой файловой системы теперь можно настроить аутентификацию **Kerberos** для поддержки безопасности на основе пользователей. Хотя конфигурация серверов **Kerberos и LDAP** выходит за рамки задач **RHCE**, для экзамена **RHCE** необходимо контролировать доступ к общим ресурсам **NFS** с помощью **Kerberos**. Глава 16 будет посвящена этой теме, а также опциям безопасности на основе хоста.

Безопасность службы SMB

SMB, указанный в целях RHCE, обозначает **сервера** по протоколу **Server Message Block**. Это сетевой протокол, первоначально разработанный **IBM**, а затем измененный **Microsoft**, как сетевой протокол для своих операционных систем. В то время как **Microsoft** теперь называет ее «Общая файловая система Интернета» (**CIFS**), реализация этого сетевого протокола в **Linux** до сих пор называется **Samba**.

Как реализовано для **RHEL** 7, вы можете использовать **Samba** для аутентификации через **Microsoft Active Directory**. **Samba** поддерживает отображение таких пользователей и групп в базу данных аутентификации **Linux**. **Samba** также поддерживает как пользовательскую, так и основанную на хосте безопасность на глобальном и общем уровнях каталогов, как описано в главе 15.

Стандартная версия **Samba** для **RHEL 7 - 4.1**. С выпуском версии 4 **Samba** также может выступать в качестве контроллера домена, совместимого с **Microsoft Active Directory**. Однако эта конфигурация выходит за рамки экзамена **RHCE**.

Безопасность службы SMTP

RHEL поддерживает две разные службы связи по электронной почте через **SMTP**-протокол: **Postfix** и **Sendmail**. Оба выпускаются по лицензиям с открытым исходным кодом.

Служба электронной почты **SMTP** по умолчанию для **RHEL** 7 - это **Postfix**, хотя вы можете настроить любую службу для достижения соответствующей цели **RHCE**. В любом случае сервис обычно прослушивает только адрес локального хоста, который является одним из уровней безопасности. Другие уровни безопасности возможны в зависимости от хостов, имен пользователей и многого другого. Для получения дополнительной информации см. Главу 13.

SSH Сервис-специфическая безопасность

Служба **SSH** устанавливается по умолчанию даже при минимальной установке **RHEL** 7. Это поощряет ее использование в качестве инструмента удаленного администрирования. Однако существуют риски, связанные с сервером **SSH**, которые можно минимизировать. Например, удаленные входы в систему с учетной записью **root** не должны быть разрешены. Безопасность может быть дополнительно отрегулирована пользователем.

Хост-безопасность

Безопасность на основе хоста относится к ограничениям доступа не только по системным именам хостов, но также по их полностью определенным доменным именам и IP-адресам. Синтаксис, связанный с безопасностью на основе хоста, может отличаться. Например, хотя каждая система распознает определенный IP-адрес, такой как **192.168.122.50**, не все службы распознают подстановочные знаки или нотацию бесклассовой междоменной маршрутизации (**CIDR**) для диапазона **IP-адресов**. В зависимости от услуги вы можете использовать одну или несколько из следующих опций для указанного диапазона сетевых адресов:

192.168.122.0/255.255.255.0
192.168.122.0/24
192.168.122.*
192.168.122.
192.168.122

Только будьте осторожны, потому что некоторые из этих опций могут привести к синтаксическим ошибкам в некоторых сетевых сервисах. Аналогичным образом, любой из следующих параметров может работать или не работать для представления всех систем в сети example.com:

***.example.com**
.example.com
example.com

Безопасность пользователя

Безопасность пользователей включает пользователей и группы. Обычно пользователи и группы, которым разрешен или запрещен доступ к услуге, собираются в список. Этот список может содержать пользователя в каждой строке, например, в файле **/etc/cron.allow**, или в списке, который следует директиве, такой как

valid users = michael donna @book

Иногда синтаксис списка пользователей жесткий; в некоторых случаях дополнительный пробел после запятой или в конце строки может привести к сбою аутентификации.

Группы часто включаются в список пользователей со специальным символом впереди, таким как **@ или +.**
Иногда пользователи, которым разрешен доступ к системе, настраиваются в отдельной базе данных аутентификации, например, связанной с сервером **Samba**, настроенной с помощью команды **smbpasswd**.

Безопасность Консоли

Как обсуждалось в главе 5, безопасность консоли управляется файлом **/etc/securetty**. Это может помочь вам отрегулировать доступ локальной консоли к корневым и обычным пользователям.

Однако консольный доступ не только локальный. Чтобы получить полное представление о безопасности консоли, вам необходимо настроить ограничения на доступ к удаленной консоли. Два основных варианта - это **SSH**, как обсуждалось ранее, и **Telnet**. Хотя команда **telnet** имеет свои применения, как описано в главе 2, связь с серверами **Telnet** по своей природе небезопасна. Имена пользователей, пароли и другие сообщения с сервера **Telnet** и с него передаются в виде открытого текста. Это означает, что анализатор сетевых протоколов, такой как **Wireshark**, может использоваться для считывания этих имен пользователей, паролей и любой другой важной информации.

Несмотря на то, что для серверов **Telnet** доступны опции на основе **Kerberos**, большинство специалистов по безопасности избегают использования **Telnet** для удаленных консолей практически любой ценой, и это соответствует рекомендациям **NSA**.

Рекомендации Агентства национальной безопасности США

АНБ проявило особый интерес к **Linux**, в частности к **Red Hat Enterprise Linux. NSA** не только потратило время на разработку **SELinux**, но и разработало руководства, которые помогут администраторам, таким как вы, создать более безопасную конфигурацию **RHEL**. (Да, «суперсекретный» **АНБ** выпустил код **SELinux** под лицензиями с открытым исходным кодом для всеобщего обозрения.) Они признают важность **Linux** в инфраструктуре компьютерных сетей. Наблюдатели **RHEL** могут заметить, как изменения между RHEL 5, RHEL 6 и RHEL 7 следуют рекомендациям АНБ.

АНБ включает в себя пять общих принципов защиты операционных систем в целом и RHEL в частности:

- ***Шифруйте передаваемые данные, когда это возможно***. Рекомендации АНБ по шифрованию включают в себя связь по частным и безопасным сетям. Использование **SSH** с параметрами безопасности, описанными в главе 11, является отличным шагом в этом процессе.
- ***Минимизируйте программное обеспечение, чтобы минимизировать уязвимость***. Согласно предложению АНБ, «Самый простой способ избежать уязвимостей в программном обеспечении - это избегать установки этого программного обеспечения». АНБ уделяет особое внимание любому программному обеспечению, которое может взаимодействовать по сети, включая графический интерфейс Linux. Минимальная установка RHEL 7 включает в себя гораздо меньше пакетов, чем аналогичная установка RHEL 5.
- ***Запускать разные сетевые сервисы в разных системах.*** Это согласуется с концепцией бастионных серверов, описанной ранее в этой главе. Внедрение упрощается благодаря гибкости, предоставляемой технологиями виртуальных машин, такими как KVM.
- ***Настройте средства безопасности для повышения надежности системы***. Задачи RHCSA и RHCE хорошо освещены с использованием межсетевых экранов на основе зон, **SELinux** и соответствующих служб сбора журналов.
- ***Используйте принцип наименьших привилегий.*** В принципе, вы должны предоставить пользователям минимальные привилегии, необходимые для выполнения их задач. Это означает не только минимизацию доступа к корневой учетной записи администратора, но и осторожное использование привилегий команды **sudo**. Опции **SELinux**, такие как роль **user_u** для ограничения (описанная в главе 4), также могут быть полезны для этой цели.

Комплект Для Разработки Политики (PolicyKit)

PolicyKit - это еще один механизм безопасности, предназначенный для защиты различных инструментов администрирования. Большинство инструментов администрирования, запускаемых в графическом интерфейсе пользователя с обычной учетной записи, запрашивают пароль администратора с окном, похожим на окно, показанное на **рисунке 10-1**.

В качестве альтернативы вы можете увидеть немного другое окно, подобное показанному на **рисунке 10-2.** Функционально эффект тот же. Как описано в окне, требуется аутентификация суперпользователя. В этом случае вам все равно придется ввести пароль администратора root.

PolicyKit хранит свои политики в каталоге **/usr/share/polkit-1/actions**. Соответствующим файлом для инструмента **system-config-date** является org.fedoraproject.config.date.policy.

РИСУНОК 10-1 Доступ к инструментам администрирования в графическом интерфейсе требует пароля root.

РИСУНОК 10-2 Доступ к административным инструментам может быть ограничен с помощью PolicyKit.

Эти файлы политики настроены в формате **XML** и могут быть изменены в дальнейшем для поддержки детального управления отдельными пользователями. Хотя **PolicyKit** предоставляет **API**, который также может использоваться текстовыми программами, он обычно используется для авторизации выполнения инструментов на основе **GUI**.

Одной из альтернатив, которая также обеспечивает детальное управление, является файл **/etc/sudoers**, описанный в главе 8.

ЦЕЛЬ СЕРТИФИКАЦИИ 10.02

Межсетевые экраны и трансляция сетевых адресов

Как правило, межсетевые экраны находятся между внутренними локальными сетями и внешними незащищенными сетями, такими как Интернет. Межсетевой экран может быть настроен для проверки каждого сетевого пакета, который проходит в или из вашей локальной сети. При настройке с соответствующими правилами он может отфильтровывать те пакеты, которые могут представлять угрозу безопасности для систем в локальной сети.

Однако, следуя духу рекомендаций **NSA**, вы должны настроить брандмауэр на каждой системе.

Хотя преобразование сетевых адресов (**NAT**) может быть реализовано в каждой системе локальной сети, оно чаще используется в тех системах, которые настроены как шлюз или маршрутизатор между локальной сетью и внешней сетью.

Определения

Брандмауэры на основе службы **firewalld** считывают заголовки каждого сетевого пакета. На основании информации, содержащейся в заголовках, вы можете настроить правила **firewalld** для фильтрации каждого пакета. Чтобы понять, как работает фильтрация пакетов, вам нужно немного понять, как информация передается по сетям.

Перед отправкой сообщения по сети это сообщение разбивается на более мелкие блоки, называемые пакетами. Административная информация, включая тип данных, адреса источника и назначения, а также порты источника и назначения (для трафика TCP и UDP), добавляется в заголовок каждого пакета. Пакеты отправляются по сети и достигают конечного хоста Linux. Брандмауэр может проверять поля в каждом заголовке. На основании существующих правил межсетевой экран может затем выполнить одно из следующих действий с этим пакетом:

- Разрешить прохождения пакета в систему.
- Переслать пакет в другие системы, если текущая система является шлюзом или маршрутизатором между сетями.
- Оцените ограничение трафика.
- Отклонить пакет с сообщением, отправленным на исходный IP-адрес.
- Отбросьте пакет без отправки какого-либо сообщения.

Каким бы ни был результат, решение может быть зарегистрировано в системном журнале или в подсистеме аудита. Если значительное количество пакетов отклонено или отброшено, файл журнала может быть полезен.

RHEL 7 поставляется со всем необходимым для настройки системы в качестве брандмауэра как для сетей **IPv4**, так и для сетей **IPv6**.

NAT может скрывать **IP**-адреса компьютеров локальной сети, которые подключаются к внешним сетям. **NAT** заменяет внутренний адрес источника **IP**-адресом интерфейса брандмауэра, подключенного к внешней сети. Внутренний адрес источника вместе с другой информацией, идентифицирующей соединение, хранится в таблице соединений брандмауэра, чтобы помнить, какой хост сделал запрос.

Когда брандмауэр получает ответ, такой как содержимое веб-страницы, процесс переворачивается. Когда пакеты проходят через брандмауэр, хост-адресат определяется в таблице соединений. Брандмауэр изменяет заголовок **IP** каждого пакета перед отправкой пакетов в пути.

Этот подход полезен по нескольким причинам. Сокрытие внутренних **IP**-адресов затрудняет хакеру «черной шляпы» проникновение во внутреннюю сеть. **NAT** поддерживает соединения между системами с частными **IP**-адресами и внешними сетями, такими как Интернет. Это причина, почему адресация **IPv4** сохранилась так долго. В мире Linux этот процесс также известен как маскировка(***masquerading.*****) IP**.

В то время как маскировка **IP** обычно упоминается как «источник(**source**) **NAT**», существует также другая форма **NAT**, которая работает в обратном направлении и известна как переадресация**(forwarding)** портов или назначение **(destination) NAT**. Переадресация портов может скрывать внутренний порт и IP-адрес службы. В качестве примера, предположим, у вас

есть внутренний сервер **с IP-адресом 192.168.122.50**, на котором запущен **веб-сервис** через **TCP-порт 8080**. С переадресацией портов вы можете подключить клиентов к другому IP-адресу и порту, например общедоступному IP-адресу на порту 80, и переслать трафик на хост и порт во внутренней сети.

Структура firewalld

ТАБЛИЦА 10-1 Элементы firewalld

Элемент зоны	Описание
Interfaces	Сетевые интерфейсы, связанные с зоной
Sources	**Исходные IP-адреса**, связанные с зоной
Services	**Входящие сервисы**, которые разрешены через зону, такую как **http**
Ports	**Порты TCP или UDP назначения**, которые разрешены через зону, например **8080/tcp**
Masquerade	Указывает, включена ли трансляция исходного сетевого адреса (**маскарадинг**)
Forward ports	Правила переадресации портов (сопоставление трафика, отправляемого на локальный порт, с другим портом того же или другого хоста)
ICMP blocks	Используется для **блокировки** сообщений **ICMP**
Rich rules	Расширенные правила брандмауэра

В главе 4 мы познакомились с некоторыми основными понятиями **firewalld**. В этом разделе мы рассмотрим его расширенные функции, такие как конфигурация зоны и расширенные правила.

Как вы уже знаете, **firewalld** основан на зонах. Зона определяет уровень доверия сетевых подключений. Основные элементы, которые определяют зону, показаны в **таблице 10-1**.
Вы можете получить список всех зон **firewalld**, введя следующую команду:

```
# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
```

Они соответствуют зонам, с которыми вы уже столкнулись в **Таблице 4-8 Главы 4**. Потратьте несколько минут, чтобы просмотреть содержимое этой таблицы.

Чтобы отобразить настройки, связанные с зоной, используйте переключатель команды **--list-all.** В качестве примера, давайте покажем все параметры конфигурации публичной зоны:

```
# firewall-cmd --list-all --zone=public
public (default, active)
interfaces: eth0
sources:
services: dhcpv6-client ssh
ports:
masquerade: no
forward-ports:
icmp-blocks:
rich-rules:
```

Как видно из вывода, публичная зона связана с интерфейсом **eth0**. Входящий трафик для служб **DHCPv6 и SSH** разрешен через зону, а маскировка отключена. Обратите внимание, что в первой строке вывода команды открытая зона помечена как «по умолчанию» и «активна».

Активная зона - это зона, связанная как минимум с одним сетевым интерфейсом или **IP**-адресом источника в **firewalld**. Мы ввели понятие зоны по умолчанию в **главе 4**: только

одна зона может быть помечена как зона «**по умолчанию**», и этот особый статус означает, что любые сетевые интерфейсы, добавленные в систему, будут автоматически назначены этой зоне.

Кроме того, зона по умолчанию действует как «ловушка для всех». Это связано с тем, как **firewalld** назначает входящие пакеты зоне, основываясь на следующих правилах:

- Если адрес источника пакета соответствует адресам источника, связанным с зоной, то пакет обрабатывается в соответствии с правилами этой зоны.
- Если пакет поступает из сетевого интерфейса, связанного с зоной, то пакет обрабатывается в соответствии с правилами этой зоны.
- В противном случае пакет обрабатывается в соответствии с правилами зоны по умолчанию.

Как только входящий пакет сопоставляется с зоной, **firewalld** обрабатывает его в соответствии с правилами этой зоны. Например, на основе предыдущего перечисленного вывода **firewall-cmd** входящие пакеты, которые достигают интерфейса **eth0**, будут обрабатываться с использованием настроек публичной**(public)** зоны **firewalld**. Согласно правилам этой зоны, трафик будет разрешен, только если он принадлежит протоколу **DHCPv6 или SSH**.

Наиболее распространенные параметры **firewall-cmd**, связанные с настройкой зоны, показаны в **таблице 10-2**. Некоторые из перечисленных опций относятся к переключателю команды **--zone** для зон не по умолчанию.

В следующем примере показано, как установить зону по умолчанию для «рабочей**(work)**» зоны:

```
# firewall-cmd --get-default-zone
public
# firewall-cmd --get-active-zones
public
        interfaces: virbr0 virbr1 wlp4s0
# firewall-cmd --set-default-zone=work
# firewall-cmd --get-default-zone
work
# firewall-cmd -get-active-zones
work
        interfaces: virbr0 virbr1 wlp4s0
```

ТАБЛИЦА 10-2 Параметры конфигурации зоны firewall-cmd

Вариант команды	Описание
--get-default-zone	Перечисляет зону по умолчанию
--set-default-zone=ZONE	Устанавливает зону по умолчанию **ZONE**
--get-zones	Список всех зон
--get-active-zones	Перечислите только активные зоны, то есть зоны, связанные по крайней мере с одним исходным интерфейсом или адресом в **firewalld**.
--list-all-zones	Перечисляет все настройки для всех зон
--list-all [--zone = ZONE]	Перечисляет все настройки для определенной зоны или зоны по умолчанию
--add-source = NETWORK [--zone = ZONE]	Связывает исходную сеть с зоной или зоной по умолчанию
--change-source = NETWORK [--zone = ZONE]	Изменяет исходную сеть, в настоящее время назначенную зоне, на другую зону или зону по умолчанию.

--remove-source = NETWORK [--zone = ZONE]	Удаляет исходную сеть из зоны или зоны по умолчанию
--add-interface = ИНТЕРФЕЙС [--zone = ZONE]	Добавляет интерфейс в ЗОНУ или в зону по умолчанию
--change-interface = ИНТЕРФЕЙС [--zone = ZONE]	Изменяет интерфейс, назначенный в данный момент зоне на другую зону или зону по умолчанию.
--remove-interface = ИНТЕРФЕЙС [--zone = ZONE]	Удаляет интерфейс из зоны или зоны по умолчанию

Обратите внимание, как все интерфейсы, которые были назначены для «публичной(**public**)» зоны, были перемещены в «рабочую(**work**)» зону после изменения. Также обратите внимание, что опция **--set-default-zone** вносит постоянное изменение, которое сохраняется после перезагрузки системы. Это одна из немногих опций в **firewalld**, для которой не требуется ключ **--permanent**, как вы увидите ниже.

Следующий пример связывает интерфейс **virbr1 с зоной «dmz»** и добавляет исходный **IP-диапазон 192.168.99.0/24 в** «публичную(**public**)» зону:

```
# firewall-cmd --change-interface=virbr1 --zone dmz
success
# firewall-cmd --add-source 192.168.99.0/24 --zone=public
success
# firewall-cmd --get-active-zones
dmz
        interfaces: virbr1
work
        interfaces: virbr0 wlp4s0
public
        sources: 192.168.99.0/24
```

Изменение конфигурации, сделанное в зоне **public** в предыдущем примере, не выдержит перезагрузки. Как было отмечено в **главе 4**, для внесения постоянных изменений в конфигурацию большинство действий **firewall-cmd** требует параметр **--permanent**. После того, как новые настройки сохранены в постоянной конфигурации, запустите **firewall-cmd --reload**, чтобы сразу применить настройки в конфигурации времени выполнения.

!!!!
Когда вы запускаете *firewall-cmd* для внесения изменений в конфигурацию, не забывайте параметр команды *--permanent*; в противном случае любые внесенные вами изменения не выживут после перезагрузки.
!!!!

Настройка сервисов и портов

Чтобы настроить эффективную зону **firewalld**, вам нужно дать ей возможность разрешать или блокировать трафик. Вы можете сделать это, внеся соответствующие изменения в службы и конфигурацию порта. Прежде чем углубляться в детали, изучите **Таблицу 10-3,** в которой представлен список параметров конфигурации службы и порта.

Двумя распространенными способами разрешения трафика через **firewalld** является добавление в зону **предварительно определенной службы или комбинации порта и протокола**, например **8080/tcp**. По умолчанию все зоны, кроме одной, содержат неявное правило «запретить весь трафик**(deny all traffic)**». Исключением является «доверенная**(trusted)**» зона, которая разрешает весь трафик по умолчанию. Следовательно, за

исключением «доверенной **(trusted)**» зоны, вы должны явно разрешить службу или порт; в противном случае соответствующий трафик будет заблокирован брандмауэром.

ТАБЛИЦА 10-3 Параметры настройки службы и порта firewall-cmd

Вариант команды	Описание
--get-services	Перечисляет все предопределенные услуги
--list-services [--zone = ZONE]	Перечисляет все службы, разрешенные для указанной зоны или зоны по умолчанию
--add-service = SERVICE [--zone = ZONE]	Разрешает трафик для указанного СЕРВИСА через ЗОНУ или зону по умолчанию
--remove-service = SERVICE [--zone = ZONE]	Удаляет СЕРВИС из ЗОНЫ или зоны по умолчанию
--list-ports [--zone = ZONE]	Перечисляет порты назначения TCP и UDP, разрешенные через ZONE или зону по умолчанию
--add-port = PORT / PROTOCOL [--zone = ZONE]	Разрешает трафик для указанного ПОРТА / ПРОТОКОЛА через ЗОНУ или зону по умолчанию
--remove-port = PORT / PROTOCOL [--zone = ZONE]	Удаляет ПОРТ / ПРОТОКОЛ из ЗОНЫ или зоны по умолчанию

!!!!

Внесение изменений в конфигурацию *firewalld* на производственном сервере может быть опасным и привести к потере административного доступа к хосту. Чтобы избежать этой проблемы, вы можете включить параметр *--timeout = SECONDS* в команду *firewall-cmd*, которая применяет изменение конфигурации только в течение указанного количества секунд.

!!!!!

Посмотрите на службы брандмауэра, определенные по умолчанию, выполнив следующую команду:

```
# firewall-cmd --get-services
amanda-client bacula bacula-client dhcp dhcpv6 dhcpv6-client dns ftp
high-availability http https imaps ipp ipp-client ipsec kerberos kpasswd
ldap ldaps libvirt libvirt-tls mdns mountd ms-wbt mysql nfs ntp openvpn
pmcd pmproxy pmwebapi pmwebapis pop3s postgresql proxy-dhcp radius rpc-bind
samba samba-client smtp ssh telnet tftp tftp-client transmission-client
vnc-server wbem-https
```

Эти службы настраиваются в файлах **XML** в каталоге **/usr/lib/firewalld/services/**. Вы можете добавить службы в каталог **/etc/firewalld/services/**.

Чтобы увидеть, как это работает, посмотрите на **рисунок 10-3**. Обратите внимание, что содержимое файла **http.xml** содержит декларацию **XML и блок <service>** с тремя дополнительными элементами: **коротким именем службы, описанием, а также соответствующим протоколом и портом, связанным с этой службой (в данном случае TCP/80).**

Просмотрите службы, связанные с зоной по умолчанию. Если вы установили по умолчанию другую зону, отмените изменения, выполнив команду **firewall-cmd --set-default-zone=public**. Далее перечислите сервисы, связанные с зоной, с помощью следующей команды:

 # firewall-cmd --list-services
dhcpv6-client ftp http ssh

Если вы завершили лабораторные работы в **главах 1 и 2** на своей физической рабочей станции, вы должны увидеть протоколы **FTP** и **HTTP** в списке служб, связанных с настройками по умолчанию зона. Если служба отсутствует, вы можете навсегда добавить ее в конфигурацию зоны, используя следующие команды:

 # firewall-cmd --permanent --add-service=ftp
 # firewall-cmd --reload

РИСУНОК 10-3. Конфигурация firewalld для службы http

```
[root@server1 ~]# cat /usr/lib/firewalld/services/http.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>WWW (HTTP)</short>
  <description>HTTP is the protocol used to serve Web pages. If you plan to make
 your Web server publicly available, enable this option. This option is not requ
ired for viewing pages locally or developing Web pages.</description>
  <port protocol="tcp" port="80"/>
</service>
[root@server1 ~]#
```

Вы также можете указать трафик, который будет разрешен через зону, используя пару **порт/протокол**. Например, предположим, что у вас **есть веб-сервер**, работающий на нестандартном порту, таком как **порт TCP 81**. Чтобы разрешить подключения к этому порту через зону **firewalld** по умолчанию, выполните следующую команду:

 # firewall-cmd --permanent --add-port=81/tcp
 # firewall-cmd –reload

Следующая команда подтверждает изменение:
 # firewall-cmd --list-ports
81/tcp

Чтобы добавить новый сервис или порт в другую зону, синтаксис этих команд такой же. Вам просто нужно указать нужную зону с помощью командного ключа **--zone**.

Когда вы запускаете службу на нестандартном порту, вам может потребоваться изменить конфигурацию метки порта **SELinux** по умолчанию. Это не требуется **для веб-сервера**, работающего через **TCP-порт 81**, но может потребоваться в других ситуациях, как вы увидите в главе 11.

Подробные правила

Добавление служб и портов в зону является наиболее распространенным способом пропуска трафика через брандмауэр. Однако в некоторых ситуациях вам может потребоваться гибкость для создания более сложных правил. Например, вы можете разрешить подключения со **всех IP-адресов** в подсети, **за исключением одного конкретного хоста**. С подробными правилами вы можете удовлетворить этот тип требований и настроить правила брандмауэра, которые соответствуют более сложной логике. Но это еще не все. Вы также можете ограничить скорость входящих подключений и регистрировать любые попытки подключения к системному журналу или службе аудита.

Общие параметры **firewall-cmd**, связанные с расширенными правилами, перечислены в **таблице 10-4**. **Подробое правило** выполняет две функции: оно определяет условия, которым должен соответствовать пакет, чтобы соответствовать правилу, и указывает действие, которое должно выполняться, если пакет соответствует.

Подробное правило использует следующий базовый формат:

rule [family=<rule_family>]
[source address=<address> [invert=true]]
[destination address=<address> [invert=true]]
service|port|protocol|icmp-block|masquerade|forward-port
[log] [audit] [accept|reject|drop]

ТАБЛИЦА 10-4 Параметры подробнго правила firewalld-cmd

Вариант команды	Описание
--list-rich-rules [--zone = ZONE]	Перечисляет все расширенные правила для указанной зоны или зоны по умолчанию
--add-rich-rule = 'RULE' [--zone = ZONE]	Добавляет расширенное правило для указанной зоны или зоны по умолчанию
--remove-rich-rule = 'RULE' [--zone = ZONE]	Удаляет расширенное правило для указанной зоны или зоны по умолчанию
--query-rich-rule = 'RULE' [--zone = ZONE]	Проверяет, было ли добавлено расширенное правило для указанной зоны или для зоны по умолчанию

Теперь давайте проанализируем эту команду, элемент за элементом. **Первый элемент - это ключевое слово правила, за которым следует необязательный тип семейства**. Существует два семейных варианта правила:

- **family = «ipv4»** Ограничивает действие правила пакетами IPv4
- **family = «ipv6»** Ограничивает действие правила пакетами IPv6

Без ключевого слова **family** правило применяется как к **пакетам IPv4**, так и к пакетам **IPv6**.

Следующие два дополнительных элемента - это адреса источника и назначения. Вы можете указать их, используя следующий формат:

- **source address=address[/mask] [invert=true] Matches all source IP addresses within the address/mask range. If you add invert=true, the rich rule applies to all but the specified address(es).**
- **destination address=address[/mask] [invert=true] Matches all destination IP addresses within the address/mask range. If you add invert=true, the rich rule applies to all but the specified address(es).**

Шаблоны пакетов могут быть более сложными. В **TCP/IP** большинство пакетов отправляются с использованием протоколов транспортного контроля **(TCP**), протокола пользовательских дейтаграмм (**UDP**) или протоколов управляющих сообщений Интернета (**ICMP**). Связанные шаблоны пакетов перечислены здесь:

- **service name=service_name All packets are checked for a specific service.**
- **port=port_number protocol=tcp|udp All packets are checked for a specific port number and protocol.**
- **icmp-block name=icmptype_name All packets are checked for a specific ICMP type. To display a list of supported ICMP types, run the command firewall-cmd --get-icmptypes.**

Опции маскарада и форвард-порта будут рассмотрены в следующих разделах главы.

Когда **rich rule** находит совпадение с шаблоном пакета, ему нужно знать, что делать с этим пакетом. Последняя часть параметров расширенного правила определяет, что происходит с сопоставленными пакетами. Есть пять основных вариантов:

- **drop The packet is dropped. No message is sent to the source host.**
- **reject The packet is dropped. An ICMP error message is sent to the source host.**
- **accept The packet is allowed through the firewall.**
- **log The packet is logged to syslog.**
- **audit The packet is logged to the audit system.**

Директива **limit value = rate/duration** используется для ограничения количества соединений и пакетов, регистрируемых за определенный промежуток времени. Например, **limit value=5/m** указывает, что в журнал будет записано или принято не более пяти сообщений журнала в минуту, **limit value=10/h** устанавливает ограничение в 10 в час и т. Д. Синтаксическая ссылка **firewalld rich language**, приведена на справочной странице **firewalld.richlanguage**.

УПРАЖНЕНИЕ 10-1

Настроить Rich Rules

В этом упражнении вы создадите расширенное правило, разрешающее **весь веб-трафик** со всех хостов в сети, кроме хоста **tester1.example.com**. Созданное вами правило запретит доступ к этому хосту с сообщением об ошибке **ICMP**. Кроме того, **узлу oustider1.example.org** должно быть разрешено подключаться к веб-серверу со всеми его попытками подключения, записанными в системный журнал, со скоростью, ограниченной двумя сообщениями в минуту. Используйте значение по умолчанию (публичная зона(**public zone**)) для всего трафика. Назначьте сегмент сети **192.168.100.0/24 зоне dmz**.

В этом упражнении предполагается, что вы установили виртуальные машины и настроили сервер **Apache** по умолчанию на своей физической рабочей станции, как описано в **лабораторных работах в главах 1 и 2**.

1. На своем физическом хосте убедитесь, что **Apache** запущен и вы можете получить доступ к домашней странице **Apache**, перейдя по следующему **URL-адресу**: **http://127.0.0.1**.
   # systemctl status httpd
   # elinks --dump http://127.0.0.1
2. Убедитесь, что для зоны по умолчанию установлена публичная (**public**) зона.
   # firewall-cmd --get-default-zone
   # firewall-cmd --set-default-zone = public
3. Перечислите настройки, связанные с зоной по умолчанию. Вы должны увидеть два виртуальных моста **virbr0 и virbr1** в списке интерфейсов.
   # firewall-cmd --list-all
4. Убедитесь, что **virbr1 связан с сетью 192.168.100.0/24**, и переместите этот интерфейс в зону **dmz**.
   # ip addr show virbr1
   # firewall-cmd --permanent --change-interface = virbr1 --zone = dmz
5. Если служба **HTTP** не разрешена в зоне по умолчанию, добавьте ее в конфигурацию **firewalld**.
   # firewall-cmd --permanent --add-service = http
6. Создайте расширенное правило для отклонения веб-соединений от **server1.example.com** (**192.168.122.50**):
   # firewall-cmd --permanent--add-rich-rule='rule family=ipv4 source address=192.168.122.50 service name=http reject'

7. Создайте расширенное правило для регистрации всех попыток подключения с сайта **outsider1.example.org (192.168.100.100)** и ограничьте скорость журналов двумя сообщениями в минуту.
   # firewall-cmd --permanent --zone=dmz --add-rich-rule='rule family=ipv4 source \ address=192.168.100.100 service name=http log limit value=2/m'
8. Перезагрузите конфигурацию брандмауэра, чтобы применить постоянные изменения к конфигурации времени выполнения.
   # firewall-cmd --reload
9. Протестируйте с сервера1 и укажите браузер **ELinks на 192.168.122.1**. Разрешено ли хосту подключаться?
   # elinks --dump http://192.168.122.1
10. Протестируйте от постороннего1 и наведите браузер **ELinks на 192.168.100.1**. Разрешено ли хосту подключаться? Видите ли вы какие-либо попытки подключения в **/var/log/messages**?
    # elinks --dump http://192.168.100.1
11. Верните конфигурацию **firewalld** к первоначальным настройкам.

Дополнительные рекомендации от АНБ

Простые межсетевые экраны часто являются наиболее безопасными. На экзамене лучше держать все как можно более простым, включая брандмауэры. Но **АНБ** пошло бы дальше. В нем содержатся рекомендации по правилам по умолчанию, ограничениям команды **ping** и блокированию подозрительных групп **IP-адресов**. К этим рекомендациям мы добавляем еще пару предложений по снижению рисков для системы. Хотя эти рекомендации выходят за рамки того, что предлагается целями **RHCE**, прочтите этот раздел. Если вам не очень удобна команда **firewall-cmd**, этот раздел может помочь. Хотя вы могли бы реализовать эти изменения с помощью опции **Rich Rules** в инструменте настройки брандмауэра, это менее эффективно, чем с **firewall-cmd**.

Вы можете протестировать любое из этих предложений в такой системе, как виртуальная машина **server1.example.com**, созданная в главе 2.

!!!!!!
Предлагаемые изменения в *firewalld* - это всего лишь рекомендации. Однако, поскольку требование «реализовать фильтрацию пакетов» является общим, полезно рассмотреть множество примеров.
!!!!!!

Отрегулируйте команду *ping*

Одна из ранних атак на различные интернет-системы включала команду **ping**. Из **Linux** можно залить другую систему с ключом **-f**. Если злоумышленник использует несколько систем, он может передавать тысячи или миллионы пакетов в секунду. Важно иметь возможность защитить систему от таких атак или ограничить их влияние, поскольку они могут препятствовать доступу других пользователей к вашим веб-сайтам и многим другим.

!!!!!
Параметр *-f* для команды *ping* был описан исключительно для указания на один из основных рисков в сети. В большинстве дистрибутивов Linux только root может указывать ключ -f. Во многих случаях запрещается запускать такую команду в чужой системе или против нее. Например, одна статья предполагает, что такое нападение может быть нарушением Закона о полиции и правосудии в Соединенном Королевстве с наказанием до 10 лет лишения свободы. Подобные законы существуют в других странах.
!!!!!!

В брандмауэре по умолчанию одно потенциально проблемное правило заключается в том, что весь трафик **ICMP** разрешен по умолчанию. Сообщения **ICMP** идут в обе стороны. Если вы запустите команду ping в удаленной системе, удаленная система ответит пакетом **ICMP Echo Reply**. Поэтому, если вы хотите ограничить сообщения ICMP, используйте следующие правила для фильтрации эхо-запросов **ICMP**:

 # firewall-cmd --add-icmp-block=echo-request

Добавьте это правило в виртуальную машину **server1.example.com** и измерьте количество пакетов, отправленных и полученных с вашего физического хоста на **server1**, с помощью команды ping **-f**. Затем сделайте то же самое после удаления блока **ICMP** и сравните свои результаты.

Блокировать подозрительные IP-адреса

Хакеры **Black Hat**, которые хотят взломать систему, могут скрыть свой IP-адрес источника. Поскольку никто не должен использовать частный или экспериментальный IPv4-адрес в общедоступном Интернете, такие адреса являются одним из способов скрытия. Следующие дополнения к **firewalld** будут отбрасывать пакеты, полученные из указанных блоков сетевых адресов **IPv4**:

 # firewall-cmd --add-rich-rule='rule family=ipv4 source address=10.0.0.0/8 drop'
 # firewall-cmd --add-rich-rule='rule family=ipv4 source address=172.16.0.0/12 drop'
 # firewall-cmd --add-rich-rule='rule family=ipv4 source address=192.168.0.0/16 drop'
 # firewall-cmd --add-rich-rule='rule family=ipv4 source address=169.254.0.0/16 drop'

Регулировать доступ к SSH

Поскольку **SSH** важен для администрирования удаленных систем, необходимы дополнительные меры для защиты этой службы. Конечно, можно **настроить нестандартный порт для связи SSH**. Такая мера может быть частью многоуровневой стратегии безопасности. Однако такие инструменты, как **nmap**, могут обнаружить использование **SSH** на таких нестандартных портах. Поэтому обычно лучше настроить конфигурацию сервера **SSH**, как описано в **главе 11**, а также правила брандмауэра, такие как следующие. Следующее расширенное правило разрешает весь трафик **SSH**, но ограничивает количество входящих соединений тремя в минуту:

 # firewall-cmd --add-rich-rule='rule service name=ssh accept limit value=3/m'

Убедитесь, что Firewalld работает

Как только нужные изменения сохранены с опцией **--permanent**, убедитесь, что **firewalld** работает с новыми правилами. Не забудьте перезагрузить конфигурацию с помощью следующей команды:

 # firewall-cmd –reload

Чтобы избежать запуска старого брандмауэра **iptables** (который был по умолчанию в **RHEL 6**), рекомендуется замаскировать соответствующие сервисные модули, как показано здесь:

 # systemctl mask iptables
 # systemctl mask ip6tables

ИП Маскарадинг

Red Hat Enterprise Linux поддерживает разновидность **NAT**, называемую **IP-маскарадингом. Маскарадинг IP** часто используется, чтобы разрешить доступ в Интернет с нескольких внутренних хостов, при этом разделяя только **один публичный IP-адрес**. **Маскарадинг IP** отображает несколько внутренних IP-адресов на этот единственный действительный внешний IP-адрес. Это помогает, потому что все общедоступные блоки адресов IPv4 уже выделены. Адреса IPv4 часто все еще доступны от третьих лиц, но за плату. Эта стоимость является еще одной причиной маскировки IP. С другой стороны, вы можете подумать, что системы в сетях IPv6 не нуждаются в маскировке, поскольку многие запрашивающие пользователи относительно легко получают свою собственную подсеть общедоступных адресов IPv6. Тем не менее, даже в сетях IPv6 маскирование может помочь обеспечить безопасность этой системы.

IP-маскарадинг - довольно простой процесс. Он реализован на шлюзе или маршрутизаторе, который по определению имеет два или более сетевых интерфейса. Один сетевой интерфейс обычно подключен к внешней сети, такой как Интернет, а второй сетевой интерфейс подключен к локальной сети. В небольшом офисе интерфейс, подключенный к внешней сети, может подключаться через внешнее устройство, такое как кабельный «модем» или **адаптер цифровой абонентской линии** (**DSL**). Следующие предположения сделаны для конфигурации:

- Общедоступный IP-адрес назначается сетевому интерфейсу, который напрямую подключен к внешней сети.
- Сетевые интерфейсы в локальной сети получают IP-адреса, связанные с одной частной сетью.
- Один сетевой интерфейс в системе шлюза или маршрутизатора получает IP-адрес в той же частной сети.
- Переадресация **IP(forwarding)** включена на маршрутизаторе или системе шлюзов, как описано далее в этой главе.
- Каждая система в локальной сети настроена с использованием частного IP-адреса маршрутизатора или шлюза в качестве адреса шлюза по умолчанию.

!EXAM
Очень важно понять, как защитить систему Red Hat Enterprise Linux от несанкционированного доступа.
!

!EXAM
Цели RHCE определяют использование firewalld для настройки трансляции сетевых адресов.
!!!!

Когда компьютеру в локальной сети требуется веб-страница в Интернете, пакеты направляются в брандмауэр. Брандмауэр заменяет исходный IP-адрес в каждом пакете публичным IP-адресом брандмауэра. Затем он присваивает пакету новый номер порта. Брандмауэр кэширует исходный IP-адрес источника и номер порта.

Когда пакет возвращается из Интернета в брандмауэр, он должен содержать номер порта. Если брандмауэр может сопоставить связанное правило с номером порта, назначенным предыдущему исходящему пакету, процесс будет обратным. Брандмауэр заменяет целевой IP-адрес и номер порта на частный IP-адрес внутреннего компьютера, а затем пересылает пакет обратно первоначальному клиенту в локальной сети.

На практике следующая команда включает маскирование. Указанная команда предполагает, что зона, напрямую подключенная к Интернету, называется «**dmz**»:

 # firewall-cmd --permanent --zone=dmz --add-masquerade

Вы также можете использовать **rich rule** для включения маскировки(**masquerading**). Это дает возможность контролировать, какие IP-адреса источника должны маскироваться:

# firewall-cmd --permanent --zone=dmz --add-rich-rule='rule family-ipv4 source \
address=192.168.0.0/24 masquerade'

В большинстве случаев частный IP-адрес сети не требуется, поскольку большинство локальных сетей, защищенных маскарадом, настроены на одну частную IP-сеть.

!!!!!
В «внешней» зоне маскировка включена по умолчанию. Присвоение сетевого интерфейса с выходом в Интернет для этой зоны автоматически маскирует всех внутренних клиентов, которые устанавливают соединение с Интернетом.
!!!!

Переадресация (Forwarding)IP

Переадресация IP чаще всего называется маршрутизацией. Маршрутизация имеет решающее значение для работы Интернета или любой IP-сети. Маршрутизаторы соединяют и облегчают связь между несколькими сетями. Когда вы настраиваете компьютер для поиска сайта во внешней сети, ему нужен адрес шлюза. Это соответствует IP-адресу маршрутизатора в локальной сети.

Маршрутизатор просматривает IP-адрес назначения каждого пакета. Если IP-адрес находится в одной из локальных сетей маршрутизатора, он направляет пакет непосредственно на соответствующий компьютер. В противном случае он отправляет пакет другому маршрутизатору ближе к конечному пункту назначения. Чтобы использовать систему **Red Hat Enterprise Linux** в качестве маршрутизатора, вы должны включить переадресацию IP в файле конфигурации **/etc/sysctl.conf**, добавив следующую строку:

net.ipv4.ip_forward = 1

Эти настройки вступают в силу при следующей перезагрузке. До этого новые настройки в **sysctl.conf** можно включить с помощью следующей команды:

 # sysctl –p

На физическом хосте, на котором работает гипервизор **KVM, переадресация IP** обычно включена по умолчанию, что можно проверить с помощью следующей команды:

 # sysctl net.ipv4.ip_forward

Инструмент настройки брандмауэра Red Hat

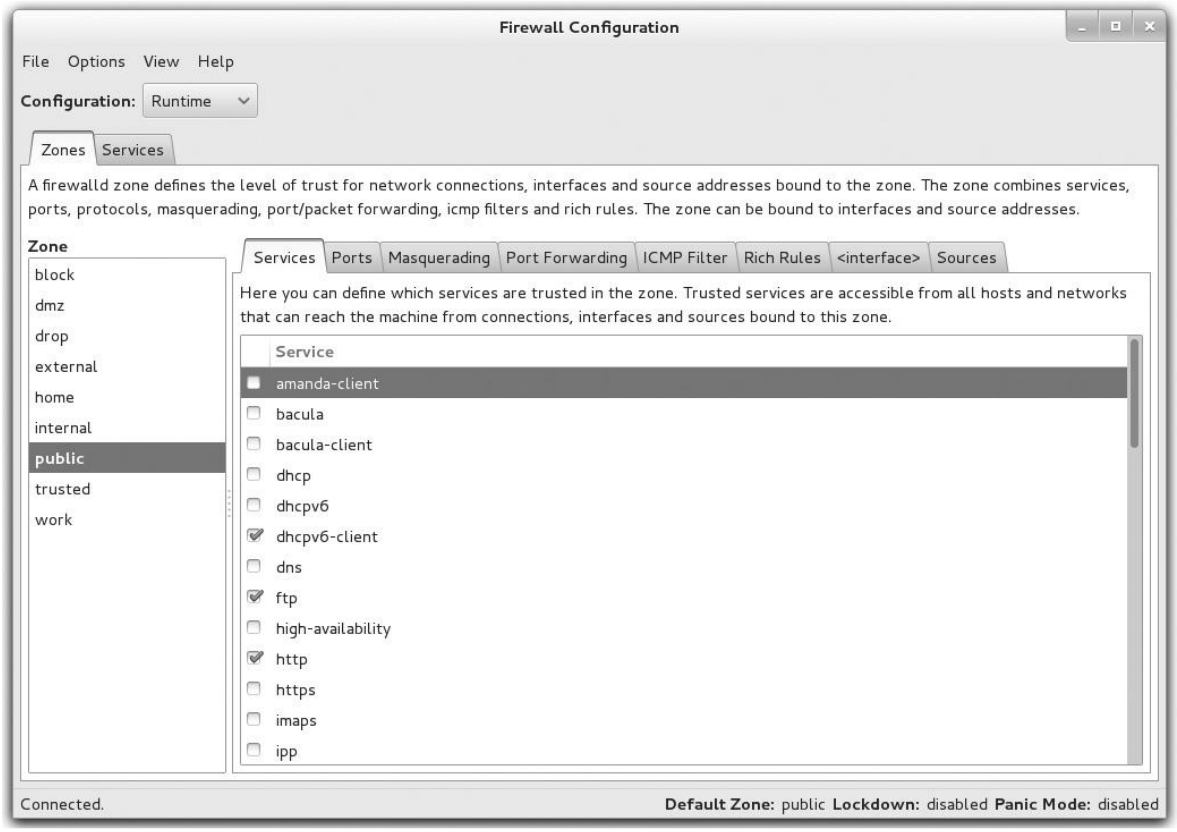
В **главе 4** были представлены элементы **RHCSA** средства настройки **Red Hat Firewall**. В этом разделе мы объясним, как вы можете реализовать цели **RHCE**, связанные с **firewalld**, используя инструмент настройки брандмауэра.

Запустите инструмент настройки брандмауэра с помощью команды **firewall-config** или нажав **Приложения | Разное | Брандмауэр**. Средство настройки брандмауэра имеет ряд возможностей, как показано на **рисунке 10-4**. В общем, если вы выбрали режим конфигурации во время выполнения (**run-time configuration mode**), он немедленно вносит изменения, но оно не сохраняются при перезагрузке системы. В большинстве случаев вы захотите выбрать

постоянный режим, который записывает изменения, которые выживают после перезагрузки системы.

После внесения изменений вы можете сразу применить сохраненную конфигурацию, нажав **Параметры | Перезагрузить Firewalld**.

РИСУНОК 10-4 Средство настройки брандмауэра



Зона по умолчанию и интерфейсы

В средстве настройки брандмауэра на левой панели перечислены зоны; зона по умолчанию выделена жирным шрифтом. В инструменте настройки брандмауэра вы можете делать все, что вы можете сделать с команда **firewall-cmd**. Вы можете назначать интерфейсы зонам, выбирать, какие сервисы и порты разрешены в зоне, включать маскировку и т. Д.

Чтобы привязать интерфейсы к зоне, нажмите вкладку «Интерфейс», чтобы открыть окно, показанное на **рисунке 10-5**. Маршрутизаторы имеют два или более сетевых интерфейса. Администраторы, которые доверяют системам во внутренней сети, могут нажать кнопку «Добавить», чтобы назначить внутренние интерфейсы «доверенной» зоне. Однако это может быть рискованной практикой. Угрозы могут исходить как снаружи, так и изнутри.

В большинстве случаев система шлюза или маршрутизатора имеет два или более устройств **Ethernet**. Предположим, вы настраиваете систему с тремя устройствами: **eth0, eth1 и eth2**, где **eth0** подключен к внешней сети**, eth1 к DMZ и eth2** к внутренней сети. Если вы доверяете всем системам в локальной сети, вы можете назначить устройство eth2 как часть доверенной зоны.

Иногда устройство может быть перечислено с использованием другого соглашения об именах. Например, беспроводной адаптер может называться wlan0 или даже ath0. В других случаях он может быть назван **wlp4s0, где p4 и s0 указывают шину PCI и номер слота соответственно**. Поэтому важно знать файлы устройств, связанные с каждым сетевым устройством в системе.

РИСУНОК 10-5 Зональные интерфейсы

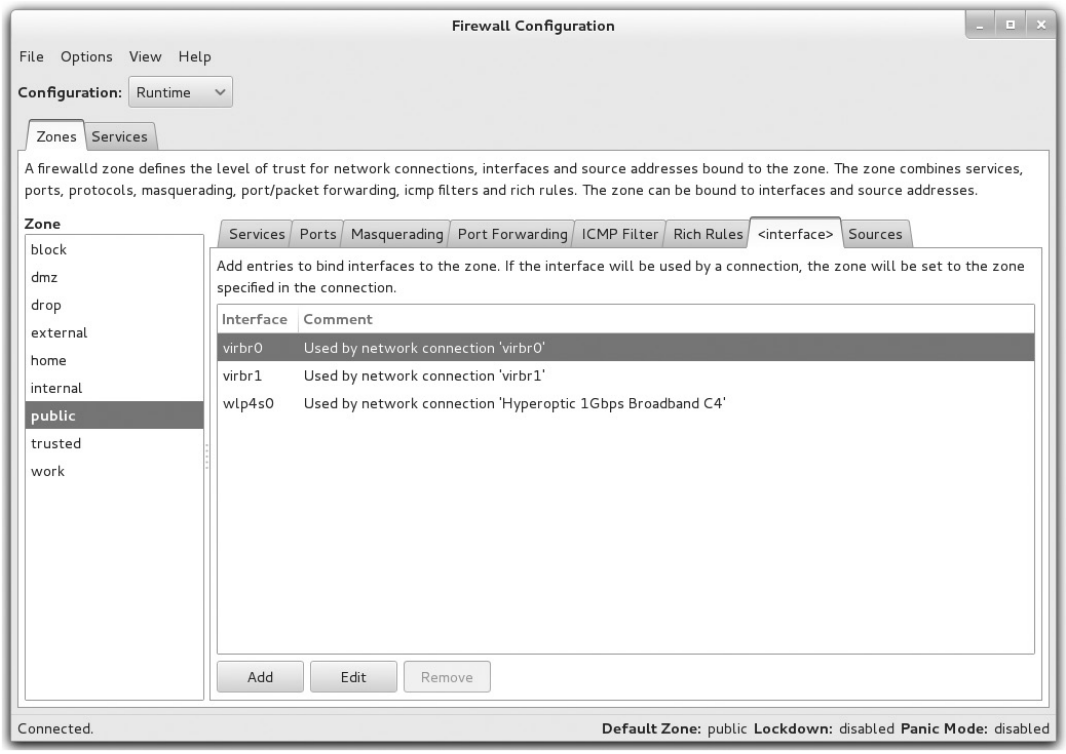
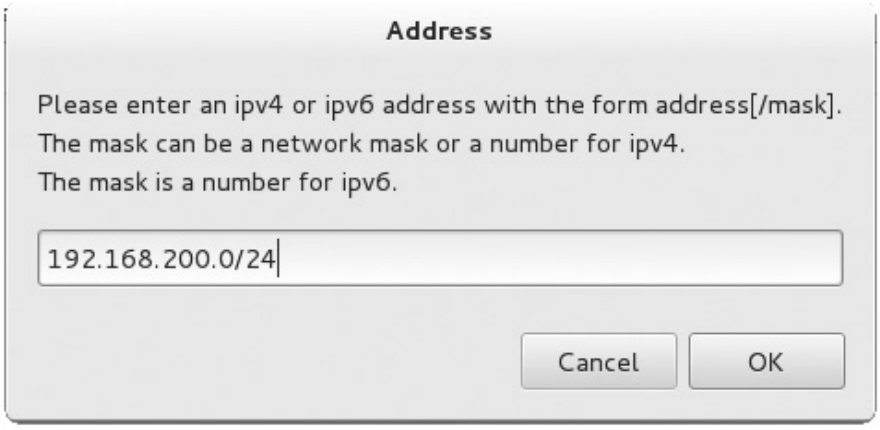


РИСУНОК 10-6
Диапазон IP-адресов источника, назначенный зоне



В средстве настройки брандмауэра вы можете привязать **исходные IP**-адреса к зоне. Для этого выберите вкладку «**Источник**» и нажмите кнопку «**Добавить**». Откроется окно адреса, показанное на **рисунке 10-6**. Затем введите диапазон IP-адресов источника, например **192.168.200.0/16**.

Точно так же вы можете включить службы и порты на соответствующих вкладках. После применения конфигурации трафик, соответствующий выбранным службам и портам, будет доверен через интерфейсы и исходные IP-адреса, связанные с зоной.

Маскировка (Masquerading)

В инструменте настройки брандмауэра выберите зону, а затем перейдите на вкладку **Маскарадинг**, чтобы открыть окно, показанное на **рис. 10-7**. В большинстве случаев вам следует настроить маскировку для трафика, выходящего в Интернет. Это имеет три преимущества:

- Он скрывает IP-адрес внутренней системы от внешних сетей.

- Требуется только один публичный IP-адрес.
- Он настраивает переадресацию IP через настроенные сетевые устройства.

Администраторы могут настроить маскировку в выбранной ими зоне. Выбранная зона должна быть подключена к внешней сети, такой как Интернет.

Перенаправление порта

В инструменте настройки брандмауэра выберите зону, а затем перейдите на вкладку «Переадресация портов **(Port Forwarding)** ». Как правило, переадресация таким способом работает только в сочетании с маскировкой. С такими правилами переадресация портов может использоваться для настройки связи путем сопоставления одного порта и протокола с портом в локальной или удаленной системе, как это определено его IP-адресом. Один пример показан на **рисунке 10-8**.

Параметры, показанные на рисунке, перенаправляют трафик, предназначенный для **TCP-порта 80** в публичной зоне, в удаленный пункт назначения **с IP-адресом 192.168.122.150**. Порт в этой удаленной системе - **8008**. Переадресацию портов также называют «NAT назначения», поскольку он используется для преобразования IP-адреса назначения и служебного порта пакета. Переадресация портов обычно используется для того, чтобы сделать внутреннюю службу видимой для других машин в Интернете, скрывая при этом все остальные службы, работающие на внутреннем хосте.

РИСУНОК 10-7 Маскарадинг с помощью инструмента настройки брандмауэра

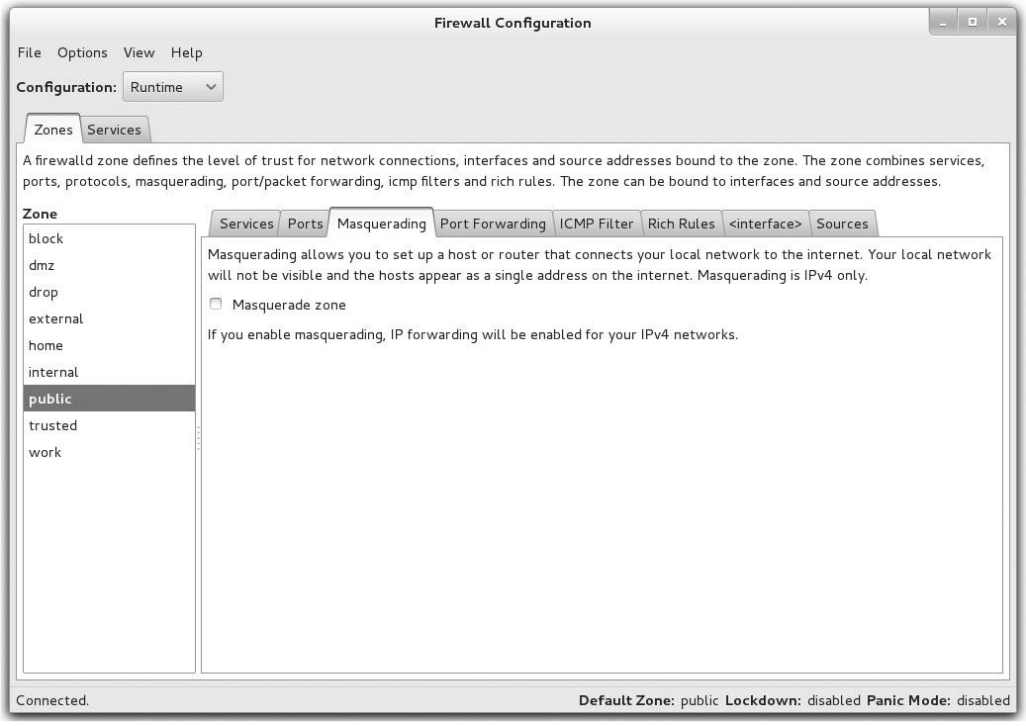
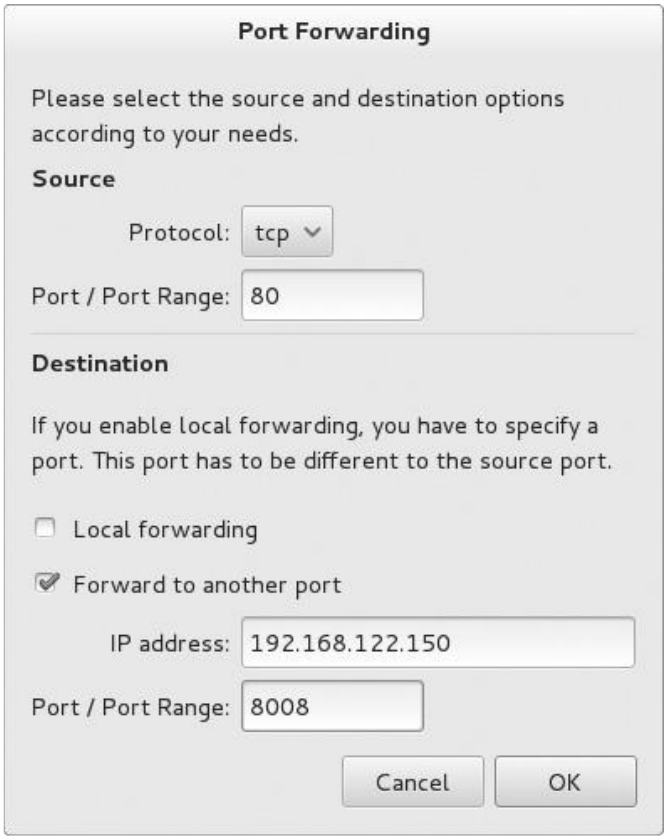


РИСУНОК 10-8
Переадресация портов с помощью инструмента настройки межсетевого экрана



ICMP фильтр

В инструменте настройки брандмауэра выберите зону и перейдите на вкладку ICMP Filter, чтобы открыть экран, показанный на **рисунке 10-9**. Как предлагается в описании, параметры относятся к различным сообщениям, связанным с протоколом **ICMP**, включая, но не ограничиваясь, связанные с **ping** пакеты. Показанные параметры дополнительно описаны в **таблице 10-5**. Если вы активируете фильтр в этой таблице, он блокирует эту категорию пакетов.

ТАБЛИЦА 10-5 Параметры фильтра ICMP

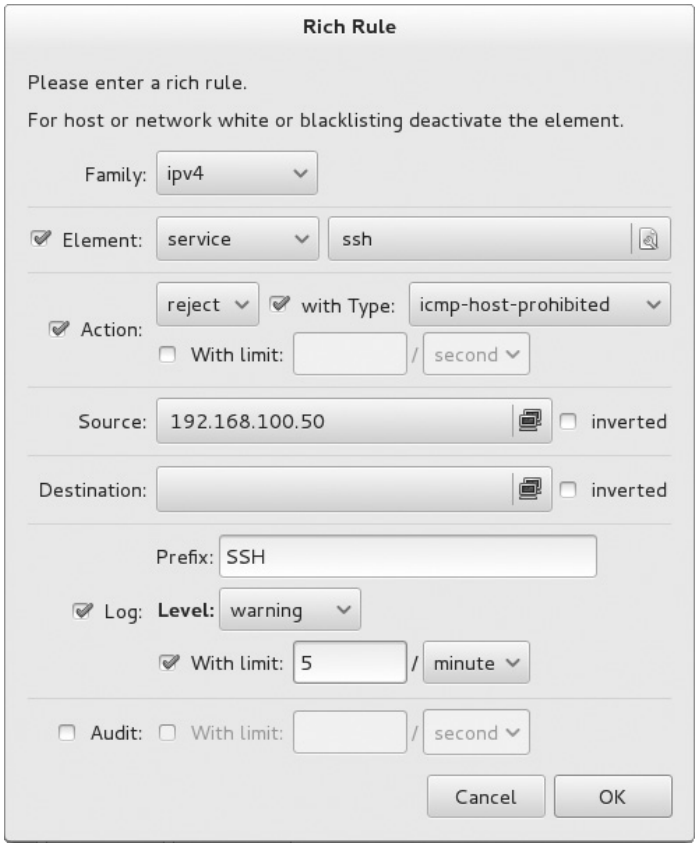
Фильтр	Описание
Destination Unreachable	Сообщение, генерируемое маршрутизатором для информирования о недоступности адреса назначения.
Echo Reply	Регулярные ответные сообщения на эхо-запросы
Echo Request	Сообщение, которое может быть сгенерировано командой ping
Parameter Problem	Сообщения об ошибках, не определенные другими сообщениями ICMP
Redirect	Сообщение, чтобы уведомить исходный хост об отправке пакетов по альтернативному маршруту
Router Advertisement	Периодическое сообщение направляется другим маршрутизаторам для объявления IP-адресов, назначенных интерфейсу.
Router Solicitation	Запрос на рекламу роутера
Source Quench	Ответ хосту для замедления передачи пакетов
Time Exceeded	Сообщение об ошибке, если в пакете превышено поле «Время жизни»

Rich Rules

Выберите зону, перейдите на вкладку **Rich Rules** и затем нажмите **Add**, чтобы открыть окно **Rich Rule**, показанное на **рисунке 10-10**.

Для целей этого раздела мы создали новое правило, как показано на **рисунке 10-10**, чтобы блокировать все соединения SSH с хоста 192.168.100.50. Попытки подключения отклоняются сообщением **ICMP Host Progged** и записываются в **/var/log/messages** с префиксом «SSH». Сообщения в журнале ограничены максимум пятью в минуту.

РИСУНОК 10-10 Воспользуйтесь преимуществами богатых правил.



ЦЕЛЬ СЕРТИФИКАЦИИ 10.03

TCP Wrappers

Как следует из его названия, **TCP Wrappers** защищает те сервисы, которые обмениваются данными по протоколу **TCP**. Первоначально он был разработан для защиты служб, настроенных с помощью демона Extended Internet Super-Server (**xinted**). Однако защита **TCP Wrappers** больше не ограничивается такими услугами; защита может применяться ко всем службам, статически и динамически связанным с файлом библиотеки-оболочки libwrap.so.0.

Способ, которым **TCP Wrappers** защищает службу, определен в **/etc/hosts.allow** и **/etc/hosts.deny** файлы конфигурации.

Защищен ли сервис оболочками TCP?

Команда **strings** может использоваться для идентификации тех демонов, которые защищены **TCP Wrappers**. Это делается путем перечисления печатных последовательностей символов, включенных в двоичный файл. Строка, связанная с **TCP Wrappers** - это **hosts_access**. Демоны можно найти в каталоге **/usr/sbin**. Таким образом, самый быстрый способ

просканировать демоны в этих каталогах на наличие строки **hosts_access** с помощью следующей команды:

 # strings -f /usr/sbin/* | grep hosts_access

Вывод зависит от установленных пакетов. Одним из примеров является демон **SSH**, **/usr/sbin/sshd.**

Вы также можете использовать команду **ldd** для общей библиотеки, чтобы подтвердить ссылку на библиотеку **TCP Wrappers, libwrap.0.so**. Чтобы определить эти зависимости для демона **sshd**, выполните следующую команду:

 # ldd /usr/sbin/sshd

Однако это не удобно, поскольку он возвращает файлы для более чем пары дюжин библиотечных файлов. Как эксперт по командной строке **Linux**, вы должны знать, как передать этот вывод команде grep, чтобы проверить, связан ли он с библиотечным файлом **TCP Wrappers, libwrap.so.0:**

 # ldd /usr/sbin/sshd | grep libwrap.so.0

И из вывода, это подтверждается:

libwrap.so.0 => /lib64/libwrap.so.0 (0x00007f13b94380000)

Файлы конфигурации **TCP Wrappers** могут помочь вам защитить **службу SSH**. Эта защита выходит за рамки любых настроек, включенных в стандартный зональный **firewalld**, файл конфигурации сервера **SSH, SELinux и так далее**. Однако такая избыточная защита важна в многоуровневой стратегии безопасности.

Файлы конфигурации TCP Wrappers

Когда система получает сетевой запрос на службу, связанную с библиотекой **libwrap.so.0**, она передает запрос в **TCP Wrappers**. Эта система регистрирует запрос и затем проверяет его правила доступа. Если нет никаких ограничений на **конкретный хост или IP-адрес, TCP Wrappers** передает управление сервису.

Ключевыми файлами являются **hosts.allow** и **hosts.deny** в каталоге **/etc**. Философия довольно проста: пользователям и клиентам, перечисленным в **hosts.allow**, разрешен доступ; пользователям и клиентам, перечисленным в **hosts.deny**, запрещен доступ. Поскольку пользователи и/или клиенты могут быть перечислены в обоих файлах, система **TCP Wrappers** выполняет следующие шаги:

1. Он ищет **/etc/hosts.allow**. Если **TCP Wrappers** находит совпадение, он предоставляет доступ. Никаких дополнительных поисков не требуется.
2. Он ищет **/etc/hosts.deny**. Если **TCP Wrappers** находит совпадение, он запрещает доступ.
3. Если хост не найден ни в одном из файлов, доступ автоматически предоставляется клиенту.

Вы используете один и тот же язык управления доступом в файлах **/etc/hosts.allow** и **/etc/hosts.deny**. Базовый формат команд в каждом файле:

daemon_list : client_list

Простейшая версия этого формата

ALL : ALL

Это определяет все службы и делает правило применимым ко всем хостам на всех IP-адресах. Если вы установите эту строку в **/etc/hosts.deny**, доступ будет запрещен для всех служб. Конечно, поскольку это читается после **/etc/hosts.allow**, сервисы в этом файле разрешены.

Вы можете создать более тонкие фильтры, чем просто запретить доступ ко всем демонам из всех систем. Например, следующая строка в /**etc/hosts.allow** позволяет клиенту с IP-адресом **192.168.122.50** подключаться к локальной системе через **Secure Shell:**

sshd : 192.168.122.50

Эта же строка в **/etc/hosts.deny** не позволит компьютеру с этим IP-адресом использовать SSH для подключения. Если в обоих файлах существует одна и та же строка, приоритет имеет **/etc/hosts.allow**, и пользователи с указанного IP-адреса смогут подключаться через **SSH**, предполагая, что другие настройки безопасности, такие как межсетевые экраны на основе зон, позволяют это. Вы можете указать клиентов несколькими различными способами, как показано в **таблице 10-6**.

ТАБЛИЦА 10-6 Образцы списков клиентов в /etc/hosts.allow и /etc/hosts.deny

Client	Description
.example.com	**Доменное имя. Поскольку это доменное имя начинается с точки, оно указывает всех клиентов в домене example.com.**
172.16.	**IP address. Поскольку этот адрес заканчивается точкой, он указывает всех клиентов с IP address of 172.16.x.y.**
172.16.72.0/255.255.254.0	**IP network address with subnet mask.**
172.16.72.0/23	**IP network address with subnet mask, but using CIDR notation.**
ALL	**Any client, any daemon.**

Как вы можете видеть в **Таблице 10-6**, есть два разных типа подстановочных знаков. **ALL** может использоваться для представления любого клиента или службы, а точка **(.)** Указывает все хосты с указанным именем домена или сетевым IP-адресом.

Вы можете настроить несколько служб и адресов запятыми. Исключения легко сделать с помощью оператора **EXCEPT**. Просмотрите следующую выдержку из файла **/etc/hosts.allow:**

ALL : .example.com
sshd : 192.168.122.0/24 EXCEPT 192.168.122.150
vsftpd : 192.168.100.100

Первая строка в этом файле открывает ВСЕ(**ALL**) службы для всех компьютеров в домене **example.com**. Следующая строка открывает службу **SSH** для любого компьютера в сети **192.168.122.0/24**, кроме компьютера с IP-адресом **192.168.122.150**. Затем служба **vsFTP** открывается на компьютер с IP-адресом **192.168.100.100**. Возможно, вы захотите добавить локальную сеть IP-адресов к отмеченным демонам в файле **/etc/hosts.allow** следующим образом:

sshd : 127. 192.168.122.0/24 EXCEPT 192.168.122.150
vsftpd : 127. 192.168.100.100

В противном случае попытки подключения из локальной системы могут быть отклонены на основании других директив в файле **/etc/hosts.deny**.

Следующая конфигурация содержит файл **hosts.deny**, чтобы увидеть, как можно создавать списки для контроля доступа:

ALL EXCEPT vsftpd : .example.org
sshd : ALL EXCEPT 192.168.122.150
ALL : ALL

В первой строке файла **hosts.deny** запрещены все службы, кроме **vsFTP**, для компьютеров в домене **example.org**. Во второй строке указывается, что единственный компьютер, которому разрешен доступ к локальному серверу **SSH**, имеет **IP-адрес 192.168.122.100**. Наконец, последняя строка - полное отрицание; всем остальным компьютерам запрещен доступ ко всем службам, контролируемым **TCP Wrappers**.

УПРАЖНЕНИЕ 10-2

Настроить TCP Wrappers

В этом упражнении вы будете использовать **TCP Wrappers** для управления доступом к сетевым ресурсам. Поскольку такие элементы управления включены по умолчанию, вам не нужно вносить какие-либо изменения в установленные службы.

1. Попробуйте подключиться к локальному **vsFTP-серверу**, используя адрес **localhost**. Возможно, вам придется сначала сделать несколько вещей:

A. Установите демон **Very Secure FTP** из **RPM**-пакета **vsftpd**.
B. Установите клиент **lftp с RPM-пакета lftp**.
C. Активируйте службу **vsFTP** с помощью команды **systemctl start vsftpd**.
D. Сконфигурируйте службу для запуска при загрузке с помощью команды **systemctl enable vsftpd.**
E. Разрешить протокол **FTP через firewalld**.

2. Отредактируйте **/etc/hosts.deny** и добавьте следующую строку (не забудьте написать файл):

   ALL : ALL

3. Что происходит, когда вы пытаетесь запустить **lftp 127.0.0.1**? А что если вы запустите команду типа **ls** после входа на **FTP-сервер**?
4. Отредактируйте **/etc/hosts.allow** и добавьте следующую строку:

   vsftpd: 127.0.0.1

5. Что происходит, когда вы пытаетесь запустить **lftp 127.0.0.1**? Команда **ls** возвращает какой-либо вывод с **FTP-сервера**?
6. Отмените любые изменения, сделанные, когда вы закончите.

ЦЕЛЬ СЕРТИФИКАЦИИ 10.04

Сменные модули аутентификации

RHEL использует систему **Pluggable Authentication Modules (PAM)** в качестве еще одного уровня безопасности, прежде всего для административных инструментов и связанных команд. **PAM** включает группу динамически загружаемых библиотечных модулей, которые определяют, как отдельные приложения проверяют своих пользователей. Вы можете изменить

файлы конфигурации **PAM**, чтобы настроить требования безопасности для различных административных утилит. Большинство файлов конфигурации **PAM** хранятся в каталоге **/etc/pam.d**.

Модули **PAM** также стандартизируют процесс аутентификации пользователя. Например, программа входа в систему использует **PAM** для запроса имен пользователей и паролей при входе в систему. Откройте файл **/etc/pam.d/login**. Посмотрите на первую строку:

auth [user_unknown=ignore success=ok ignore=ignore default=bad] pam_securetty.so

Для интерпретации эта строка означает, что пользователь root может войти в систему только с защищенных терминалов, как определено в файле **/etc/securetty**, а неизвестные пользователи игнорируются.

Файлы конфигурации, показанные в каталоге **/etc/pam.d**, часто имеют то же имя, что и команда, запускающая административную утилиту. Эти утилиты «осведомлены о PAM(**PAM aware**». Другими словами, вы можете изменить способ проверки пользователей для таких приложений, как консольная программа входа в систему. Просто измените соответствующий файл конфигурации в каталоге **/etc/pam.d**.

Конфигурационные файлы

Посмотрите файлы конфигурации в типичном каталоге **/etc/pam.d**, как показано на **рисунке 10-11**. В зависимости от того, что установлено, вы можете увидеть несколько иной список файлов.

Как предлагалось ранее, большинство имен файлов в каталоге **/etc/pam.d** являются описательными. Посмотрите на некоторые из этих файлов. В большинстве случаев они относятся к модулям **PAM**. Эти модули можно найти в каталоге **/usr/lib64/security**. Отличное описание каждого модуля можно найти в каталоге **/usr/share/doc/pam-versionnumber**, в подкаталогах **txts/** и **html/.** Например, функциональность модуля **pam_securetty.so** описана в файле **README.pam_securetty**.

Вы также можете обратиться к **HTML-версии** Руководства системного администратора **Linux-PAM**, доступной в каталоге **/usr/share/doc/pam-versionnumber/html**, начиная с файла **Linux-PAM_SAG.html**.

РИСУНОК 10-11 Файлы конфигурации PAM в /etc/pam.d каталог

```
[root@server1 ~]# \ls /etc/pam.d
atd                     login             smtp
authconfig              newrole           smtp.postfix
authconfig-gtk          other             sshd
authconfig-tui          passwd            su
chfn                    password-auth     subscription-manager
chsh                    password-auth-ac  subscription-manager-gui
config-util             pluto             sudo
crond                   polkit-1          sudo-i
cups                    postlogin         su-l
fingerprint-auth        postlogin-ac      system-auth
fingerprint-auth-ac     ppp               system-auth-ac
gdm-autologin           remote            system-config-authentication
gdm-fingerprint         rhn_register      system-config-language
gdm-launch-environment  runuser           systemd-user
gdm-password            runuser-l         vlock
gdm-pin                 setup             vmtoolsd
gdm-smartcard           smartcard-auth    vsftpd
liveinst                smartcard-auth-ac xserver
[root@server1 ~]#
```

Система **PAM** предоставляет четыре различных типа услуг. Они связаны с четырьмя различными типами правил **PAM**:

- **Authentication management (auth)** Проверяет личность пользователя. Например, правило аутентификации **PAM** проверяет, предоставил ли пользователь действительное имя **пользователя и пароль**.
- **Account management (account)** Разрешает или запрещает доступ в соответствии с политикой учетной записи. Например, правило учетной записи **PAM** может запрещать доступ в зависимости от времени, истечения срока действия пароля или определенного списка ограниченных пользователей.
- **Password management (password)** Управляет политиками смены пароля. Например, правило пароля **PAM** может применять минимальную длину пароля, когда пользователь пытается изменить свой пароль.
- **Session management (session)** Применяет настройки для сеанса приложения. Например, правило сеанса PAM может устанавливать параметры по умолчанию для консоли входа в систему.

Конфигурация, показанная на **рисунке 10-12**, взята из примера файла конфигурации **PAM, /etc/pam.d/login**. Каждая строка во всех файлах конфигурации **PAM** записана в следующем формате:

type control_flag module_name [arguments]

Тип(**type**), как описано ранее, может быть **auth, account, password или session**. **Control_flag** определяет, что делает **PAM**, если модуль преуспевает или терпит неудачу.

РИСУНОК 10-12. Файл конфигурации PAM /etc/pam.d/login

```
#%PAM-1.0
auth [user_unknown=ignore success=ok ignore=ignore default=bad] pam_securetty.so
auth       substack     system-auth
auth       include      postlogin
account    required     pam_nologin.so
account    include      system-auth
password   include      system-auth
# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
session    required     pam_loginuid.so
session    optional     pam_console.so
# pam_selinux.so open should only be followed by sessions to be executed in the
user context
session    required     pam_selinux.so open
session    required     pam_namespace.so
session    optional     pam_keyinit.so force revoke
session    include      system-auth
session    include      postlogin
-session   optional     pam_ck_connector.so
~
~
~
~
"/etc/pam.d/login" 18L, 796C
```

module_name указывает имя фактического файла модуля **PAM**. Наконец, вы можете указать параметры для каждого модуля.

Поле **control_flag** требует дополнительного пояснения. Он определяет, как **PAM** реагирует, когда модуль возвращает успех или неудачу. Пять наиболее распространенных контрольных флагов описаны в **таблице 10-7.**

ТАБЛИЦА 10-7 PAM Control Flags

control_flag	Описание
required	Если этот модуль возвращает успех, **PAM** переходит к следующему правилу этого типа. Если это не удается, **PAM** переходит к следующему правилу в файле конфигурации, но окончательный результат - сбой.
requisite	Если этот модуль дает сбой, **PAM** не проверяет никаких дополнительных правил и возвращает ошибку.
sufficient	Если этот модуль проходит, никакие другие правила этого типа не должны быть обработаны, и результатом является успех. В противном случае, если проверка не пройдена, **PAM** продолжает обрабатывать оставшиеся правила.
optional	**PAM** игнорирует успех или неудачу этого правила.
include	Включает все директивы одного типа из указанного файла конфигурации; например, если директива **password** включает в себя **system-auth**, это включает в себя все директивы пароля из файла **PAM system-auth**.

Чтобы увидеть, как работают управляющие флаги, взгляните на правила из файла конфигурации **/etc/pam.d/runuser**:

auth sufficient pam_rootok.so

Первая команда **auth** проверяет модуль **pam_rootok.so**. Другими словами, если пользователь **root** пытается выполнить команду **runuser**, правило вернет пропустить, и команда **runuser** будет выполнена. Поскольку **control_flag** установлен **sufficient**, то если бы в этом файле были другие команды **auth**, они были бы проигнорированы.

session optional pam_keyinit.so ignore

Цель второй строки - очистить кольцо сеансовых ключей процесса **runuser**, когда он завершится. В этом случае **control_flag** является **optional**, а это означает, что результат этого правила не влияет на другие правила сеанса.

session required pam_limits.so

Третья строка устанавливает ограничения ресурсов, определенные в **/etc/security/limits.conf** при вызове экземпляра приложения **runuser**. Установлен **control_flag required**, что приведет к сбою сеанса команды, если ограничения не могут быть установлены.

session required pam_unix.so

Модуль, связанный с последним типом сеанса (**pam_unix.so**), регистрирует имя пользователя и тип службы в начале и конце каждого сеанса команды.

Формат файла PAM

Этот раздел немного сложеный. Он начинается с файла конфигурации **/etc/pam.d/login**, показанного на **рисунке 10-12**. Кроме того, поскольку файл содержит ссылки на файл конфигурации **/etc/pam.d/system-auth**, показанный на **рис. 10-13**, вам нужно будет переходить назад и вперед между файлами, чтобы следовать этому разделу.

Вам не нужно запоминать содержимое этого раздела. Вместо этого вы должны использовать его, чтобы лучше ознакомиться с файлами конфигурации **PAM**. Читая этот раздел, ознакомьтесь с каждым модулем **PAM**, прочитав соответствующую справочную страницу. Команда **rpm -qd pam** выдает полный список установленных **man-страниц PAM** в вашей системе.

Когда пользователь открывает текстовую консоль и входит в систему, **Linux** построчно просматривает файл конфигурации **/etc/pam.d/login**. Как отмечалось ранее, первая строка в **/etc/pam.d/login** ограничивает доступ пользователя root к защищенным терминалам, как определено в файле **/etc/securetty:**

auth [user_unknown=ignore success=ok ignore=ignore default=bad] pam_securetty.so

РИСУНОК 10-13. Файл конфигурации PAM /etc/pam.d/system-auth

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 1000 quiet_success
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 1000 quiet
account     required      pam_permit.so

password    requisite     pam_pwquality.so try_first_pass local_users_only retry
=3 authtok_type=
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass use_au
thtok
password    required      pam_deny.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
-session     optional      pam_systemd.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet
use_uid
session     required      pam_unix.so
~
"/etc/pam.d/system-auth" 22L, 974C
```

Следующая строка содержит команды **auth** из файла конфигурации **PAM system-auth**:

auth substack system-auth

В этом примере вы можете предположить, что управляющий флаг **substack** эквивалентен директиве **include**. Файл конфигурации **system-auth**, показанный на **рисунке 10-13**, содержит четыре директивы **auth**. В вашей системе вы можете увидеть дополнительные строки - например, если вы настроили машину как клиент **LDAP** или **Kerberos**. В этом случае ваша конфигурация будет ссылаться на дополнительные модули **PAM**, такие как **pam_ldap** или **pam_krb5**.

auth required pam_env.so
auth sufficient pam_unix.so nullok try_first_pass
auth requisite pam_succeed_if.so uid >= 1000 quiet_success
auth required pam_deny.so

По порядку в предыдущих строках устанавливаются переменные среды и проверяется проверка подлинности пароля по локальным базам данных **/etc/passwd и /etc/shadow (pam_unix.so).** Флаг **sufficient**, связанный со вторым модулем, означает, что аутентификация завершается успешно, если был введен действительный пароль, и никакие дальнейшие правила из раздела **auth** не обрабатываются.

Если модуль **pam_unix.so** возвращает ошибку, **PAM** обработает следующее правило, которое отключит ведение журнала, если идентификатор пользователя учетной записи - 1000 и

выше. Затем, если **PAM** доберется до последнего правила, пользователю будет отказано в доступе (**pam_deny.so**).

Теперь вернитесь в файл **/etc/pam.d/login**. Следующая строка включает директиву в файле **postlogin**, которая не содержит никаких правил авторизации. Переходя к следующей строке, это вызывает тип учетной записи модуля **pam_login.so**. Этот модуль отключает вход пользователя, если существует файл **/etc/nologin:**

!EXAM
Если файл /etc/nologin существует, обычные пользователи не могут войти в локальную консоль. Любой обычный пользователь, который пытается войти в систему, может прочитать содержимое файла /etc/nologin в виде сообщения. Это поведение контролируется
модуль pam_nologin.so.
!!!!!

```
account        required        pam_nologin.so
```

Следующее правило включает правила учетной записи из файла конфигурации **/etc/pam.d/system-auth**:

```
account        include         system-auth
```

Это строки правил типа учетной записи из файла **/etc/pam.d/system-auth** по умолчанию:

```
account required pam_unix.so
account sufficient pam_localuser.so
account sufficient pam_succeed_if.so uid < 1000 quiet
account required pam_permit.so
```

Первая строка ссылается на модуль **pam_unix.so** в каталоге **/usr/lib64/security**, который проверяет в **/etc/shadow**, является ли учетная запись действительной и срок ее действия не истек. На основе модуля **pam_localuser.so** и на достаточный тип элемента управления, если имя пользователя указано в **/etc/passwd**, дальнейшие директивы не обрабатываются. Модуль **pam_succeed_if.so** отключает ведение журнала для пользователей службы (с идентификаторами пользователей менее 1000). Затем модуль **pam_permit.so** всегда возвращает успех.

Теперь вернитесь в файл **/etc/pam.d/login**. Следующая строка - это директива пароля, которая включает в себя другие правила пароля из файла **/etc/pam.d/system-auth**:

```
password       include         system-auth
```

Это правила типа пароля из файла по умолчанию **/etc/pam.d/system-auth**:

```
password requisite pam_pwquality.so try_first_pass local_users_only retry=3 authok_type= \
password sufficient pam_unix.so sha512 shadow nullok try_first_pass use_authok
password required pam_deny.so
```

Первое правило из этого вывода выполняет проверку надежности пароля. Он позволяет использовать пароль, собранный приложением, которое вызывает **PAM (try_first_pass)**, и применяет свои проверки только к локальным пользователям с максимум тремя попытками смены пароля.

Следующее правило обновляет пароль пользователя с использованием хэша шифрования SHA512, поддерживает набор теневых паролей, **описанный в главе 8**, позволяет использовать существующий нулевой (нулевой длины) пароль и вынуждает модуль установить новый пароль в соответствии с предоставленным значением предыдущим модулем (**use_authok**).

Директива **password required pam_deny.so** тривиальна; как отмечено в файле **README.pam_deny** в каталоге **/usr/share/doc/pam-versionlevel/txt**, этот модуль всегда дает сбой.

Наконец, в файле по умолчанию **/etc/pam.d/login** есть шесть правил сеанса. Давайте возьмем их по три за раз:

session required pam_selinux.so open
session required pam_namespace.so
session optional pam_keyinit.so force revoke

Первая строка (**pam_selinux.so open**) устанавливает несколько контекстов безопасности **SELinux**. Вторая строка **(pam_namespace.so**) создает отдельные пространства имен для пользователей при входе в систему. Третья строка инициализирует список ключей сеанса входа в систему (**pam_keyring.so**).

session include system-auth
session include postlogin
-session optional pam_ck_connector.so

Забегая вперед, последнее из этой группы правил регистрирует сеанс входа в систему с демоном **ConsoleKit**. Обратите внимание на знак минус перед строкой: это говорит **PAM** не отправлять сообщение об ошибке в системный журнал, если модуль отсутствует. Предыдущие правила включают следующие строки типа сеанса из файлов **system-auth и postlogin**:

session optional pam_keyinit.so revoke
session required pam_limits.so
-session optional pam_systemd.so
session [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session required pam_unix.so

Первое из этих правил идентично строке в главном файле **/etc/pam.d/login**, которая отменяет набор ключей сеанса вызывающего процесса. Следующее правило устанавливает ограничения (**pam_limits.so**) для отдельных пользовательских ресурсов через **/etc/security/limits.conf**. Следующее правило регистрирует сеанс пользователя с помощью менеджера входа в систему **systemd**. Четвертое правило пропустит следующее правило **(success = 1) для заданий cron**. Последнее правило регистрирует результат, когда пользователь входит в систему.

Наконец, следующие три правила, включенные в файл **postlogin**, вызывают модуль **pam_lastlogin.so** с различными параметрами в зависимости от того, является ли запрашивающая служба графическим входом в систему, командой **su** или другим процессом. Модуль **pam_lastlogin.so** показывает количество предыдущих неудачных попыток входа в систему для пользователя, а также обычно используется для записи даты последнего входа в систему в **/var/log/lastlog**.

session [success=1 default=ignore] pam_succeed_if.so service !~ gdm* service !~ su* quiet
session [default=1] pam_lastlog.so nowtmp showfailed
session optional pam_lastlog.so silent noupdate showfailed

УПРАЖНЕНИЕ 10-3

Настройте PAM для ограничения корневого доступа
В этом упражнении вы можете поэкспериментировать с некоторыми функциями безопасности PAM в Red Hat Enterprise Linux 7.

1. Сделайте резервную копию **/etc/securetty** с помощью следующей команды:

 # cp /etc/securetty /etc/securetty.bak

2. Отредактируйте **/etc/securetty** и удалите строки от **tty3 до tty11**. Сохраните изменения и выйдите.
3. Используйте **alt-f3 (ctrl-alt-f3, если вы используете X Window),** чтобы переключиться на виртуальную консоль № 3. Попробуйте войти в систему как пользователь **root**. Что происходит?
4. Повторите шаг 3 для обычного пользователя. Что происходит? Вы знаешь почему?
5. Используйте **alt-f2** для переключения на виртуальную консоль № 2 и попробуйте войти в систему как **root**.
6. Просмотрите сообщения в **/var/log/secure**. Видите ли вы, где вы пытались войти в систему как **root** в виртуальной консоли № 3?
7. Восстановите исходный файл **/etc/securetty** с помощью следующей команды:

 # mv /etc/securetty.bak /etc/securetty

Следует помнить, что файл **/etc/securetty** управляет консолями, с которых вы можете войти в систему как пользователь **root**. Следовательно, внесенные изменения не влияют на обычных (**non-root**) пользователей.

PAM и пользовательская безопасность

В этом разделе вы узнаете, как настроить **PAM** для ограничения доступа для определенных пользователей. Ключом к этой функции безопасности является модуль **pam_listfile.so**, который находится в каталоге **/usr/lib64/security**. Если вы установили сервер **vsFTP**, файл **/etc/pam.d/vsftpd** содержит пример этого модуля.

Во-первых, следующая строка в файле **vsftpd** инициализирует и очищает любые существующие **keyring**(средство управления и хранения ключей в ядре) при закрытии сеанса:

session optional pam_keyinit.so force revoke

Способ, которым **PAM** может ограничить доступ пользователя, показан в следующем правиле:

auth required pam_listfile.so item=user sense=deny file=/etc/vsftpd/ftpusers onerr=succeed

Чтобы понять, как это работает, давайте разберем это правило на его составные части. Вы уже знаете первые три части правила из предыдущего раздела. Показанные параметры связаны с модулем **pam_listfile.so**, как описано на справочной странице **pam_listfile** и в **таблице 10-8.**

Таким образом, для указанного правила (**onerr = successed**) ошибка, как ни странно, возвращает **success (item = user).** Если пользователь находится в указанном списке (**file=/etc/vsftpd/ftpusers**), правило позволяет этому пользователю (**sense = allow**) получить доступ к указанному инструменту.

!EXAM
Убедитесь, что вы понимаете, как Red Hat Enterprise Linux обрабатывает авторизацию пользователей через файлы конфигурации /etc/pam.d. Когда вы тестируете эти файлы, убедитесь, что вы создали резервную копию всего в PAM, прежде чем вносить какие-либо изменения, потому что любые ошибки, которые вы делаете в файле конфигурации PAM, могут полностью отключить доступ к вашей системе (PAM безопасен).
!!!!!

ТАБЛИЦА 10-8 Параметры для модуля pam_listfile.so

pam_listfile Option	Описание
item	Эта опция может использоваться для ограничения доступа к терминалу (**tty**), пользователю (пользователю), группе (группе) или многим другим.(**user (user), group (group), or more**)
sense	Если элемент найден в указанном файле, выполните указанное действие. Например, если пользователь указан в **/etc/special** и **sense=allow**, эта команда предоставляет пользователю разрешение для указанного инструмента.
file	Путь к файлу со списком пользователей, групп и т.д., Например, **file=/etc/special**.
onerr	Если есть проблема, сообщите модулю, что делать. Возможны следующие варианты: **onerr = success или onerr = fail**.

УПРАЖНЕНИЕ 10-4

Используйте PAM для ограничения доступа пользователей

Вы также можете использовать систему **PAM** для ограничения доступа всем пользователям без полномочий **root**. В этом упражнении вы ограничите доступ с помощью модуля **pam_nologin.so**. Он должен работать рука об руку с файлом конфигурации по умолчанию **/etc/pam.d/login**, а именно со следующей строкой:

account required pam_nologin.so

1. Найдите файл **/etc/nologin**. Если он еще не существует, создайте его с таким сообщением, как:

   Извините, доступ ограничен пользователем root
2. Получите доступ к другому терминалу с помощью команды, такой как **ctrl-alt-f2**. Попробуйте войти в систему как обычный пользователь. Что ты видишь?
3. Войдите в систему как пользователь **root**. Вы увидите то же сообщение; но как пользователь **root**, вам разрешен доступ.
4. Проверьте файл **/var/log/secure**. Ваша система отклонила попытку входа от обычного пользователя? Каковы были связанные сообщения для пользователя **root**?

СЦЕНАРИЙ И РЕШЕНИЕ	
У вас есть только один **общедоступный IP-адрес**, но вам необходимо обеспечить доступ в Интернет ко всем системам вашей локальной сети. Каждый компьютер в локальной сети имеет свой собственный **IP-адрес.**	Использование **firewalld** для маскировки **IP-адресов**. Убедитесь, что **IP-переадресация активна.**
Вы установили **SSH-сервер** в корпоративной сети и хотите ограничить доступ к определенным отделам. Каждый отдел имеет свою собственную подсеть.	Используйте файл **/etc/hosts.deny** в пакете **tcp_wrappers**, чтобы заблокировать **SSH-доступ** к нежелательным подсетям. Лучшая альтернатива - использовать **/etc/hosts.allow** для поддержки доступа к нужным отделам, а затем использовать **/etc/hosts.deny**, чтобы запретить доступ всем остальным. Подобные опции возможны при использовании **firewalld rich rules**.

Вы хотите ограничить доступ к службе, такой как **SSH**, только для определенных пользователей.	Добавьте строку в соответствующие файлы конфигурации **Pluggable Authentication Module** в файле **/etc/pam.d**, чтобы использовать модуль **pam_listfile.so**.
Вы хотите изменить локальный брандмауэр для защиты от **ICMP**-атак, таких как команда **ping flood**.	Модифицируйте брандмауэр, чтобы отклонять или запрещать определенные типы **ICMP-пакетов**.

ЦЕЛЬ СЕРТИФИКАЦИИ 10.05

Безопасные файлы и многое другое с GPG2

Учитывая важность сетевой безопасности, вы должны знать, как шифровать файлы для безопасной передачи. Компьютерный стандарт для служб шифрования и подписи файлов известен как **Pretty Good Privacy (PGP).** Реализация **PGP** с открытым исходным кодом известна как **GNU Privacy Guard (GPG).** Версия, выпущенная для **RHEL** 7, является более продвинутой и функциональной, задокументированной как **GPG версии 2 (GPG2)**. Вы, вероятно, уже использовали **GPG2** для проверки подлинности пакетов **RPM**, как обсуждалось в **главе 7**. В этом разделе такие проверки выполняются еще на один шаг; вы будете генерировать закрытые и открытые ключи, а затем использовать эти ключи для шифрования и дешифрования выбранных файлов.

Хотя **GPG** не указан в целях **RHCE**, это тема безопасности, совместимая с другими целями безопасности, обсуждаемыми в этой книге. Мы считаем, что это отличная тема, которая может быть включена в будущие версии экзамена **RHCE**.

Команды GPG2

Версия 2 **GPG**, включенная в **RHEL** 7, имеет более модульный подход к шифрованию и аутентификации. Существует даже связанный пакет, используемый для аутентификации с помощью смарт-карты. Но это не главное для новых команд **gpg2**. Доступные команды **GPG** кратко описаны в **таблице 10-9.**

Таблица 10-9 предназначена только для описания диапазона возможностей, связанных с пакетами **RHEL 7 GPG2**. Основное внимание в этом разделе уделяется шифрованию и дешифрованию файлов.

Команда	Описание
gpg	Символьная ссылка на команду **gpg2**
gpg2	Инструмент шифрования и подписи **GPG2**
gpg-agent	Демон управления ключами **GPG2**
gpgconf	Предоставляет доступ и изменяет файлы конфигурации в **~/.gnupg**
gpg-connect-agent	Утилита для связи с активным агентом **GPG2**
gpg-error	Команда для интерпретации номера ошибки **GPG2**
gpgsplit	Команда для разделения сообщения **GPG2** на пакеты
gpgv	Символьная ссылка на команду **gpg2v**
gpgv2	Команда для проверки подписи **GPG**; требуется файл подписи
gpg-zip	Команда для шифрования или подписи файлов в архив

Текущая конфигурация GPG2

Хотя страница Хотя страница **man** для команды **gpgconf** предполагает, что она просто используется для изменения каталога с помощью связанных файлов конфигурации, эта команда делает больше. Особняком по умолчанию это переключатель **--list-components**, который

указывает полный путь к соответствующим исполняемым файлам. С ключом **--check-programs** он обеспечивает выполнение всех связанных программ. **gpgconf** также можно использовать для проверки синтаксиса файла конфигурации **GPG2**. Типичный вариант - в домашнем каталоге текущего пользователя в подкаталоге **.gnupg/**. Другой типичный параметр находится в каталоге **/etc/gnupg**.

Параметры шифрования GPG2

Генерация ключа **GPG2** включает в себя выбор из трех различных криптографических алгоритмов, как указано далее. Каждый из этих алгоритмов включает в себя открытый и закрытый ключи. Открытый ключ может быть передан другим для использования при шифровании файлов и сообщений. Закрытый ключ используется владельцем и является единственным способом расшифровки файла или сообщения.

- **RSA** Назван в честь разработчиков **Rivest, Shamir и Adleman**. Хотя типичные ключи **RSA** имеют длину 1024 или 2048 бит, они могут быть больше. Более короткие ключи в 512 бит были взломаны. **RSA** находится в свободном доступе.
- **DSA** Алгоритм цифровой подписи. Предложенный Национальным институтом науки и технологии США, DSA был доступен для использования по всему миру, без лицензионных отчислений. Это государственный стандарт США, в котором в качестве хеш-функций дайджеста сообщений используются версии **SHA-1 и SHA-2** алгоритма безопасного хеширования (**SHA**). **SHA-1** постепенно сокращается; **SHA-2** включает в себя шесть хеш-функций с дайджестами сообщений до **512 бит**, также известными как **SHA-512**, тот же хеш, который сейчас используется для набора теневых паролей **RHEL 7**.
- **ElGamal** Разработанный **Taher Elgamal**, этот вероятностный алгоритм шифрования используется в **GPG** в сочетании с **DSA**, при этом пара ключей **ElGamal** используется для шифрования, а другая пара ключей **DSA** - для создания подписей. **ElGamal** - это первая схема шифрования, основанная на методе обмена ключами **Диффи-Хеллмана**.

Создать ключ GPG2

Команда **gpg --gen-key** может использоваться для настройки пар ключей с различными типами схем шифрования. Перед запуском команды подготовьтесь с ответами на следующие вопросы:

- Количество битов для ключей шифрования. Обычно максимальное количество битов составляет 4096, но сложный ключ шифрования может занять несколько минут.
- Желаемое время жизни ключей. Особенно, если вы устанавливаете ключи с меньшим количеством битов, вы должны предположить, что решительный хакер в черной шляпе сможет расшифровать ключ в течение некоторого количества месяцев или недель.
- Имя, адрес электронной почты и комментарий. Хотя имя и адрес электронной почты не обязательно должны быть реальными, другие будут воспринимать их как часть открытого ключа.
- Ключевая фраза. Хорошие парольные фразы должны включать пробелы, строчные и прописные буквы, цифры и знаки препинания.

Как указано, команда **gpg --gen-key** запрашивает одну из четырех различных схем шифрования. Как следует из метки (только подписи), связанной с вариантами 3 и 4, эти опции работают просто как цифровые подписи, а не для шифрования.

Please select what kind of key you want:
(1) RSA and RSA (default)
(2) DSA and Elgamal

(3) DSA (sign only)
(4) RSA (sign only)
Your selection?

Все четыре варианта следуют аналогичной последовательности шагов. Например, если вы выберете опцию 2, появится следующий вывод:

DSA keys may be between 1024 and 3072 bits long.
What keysize do you want? (2048)

По умолчанию установлено значение 2048 бит, которое выбирается, если вы просто нажмете ввод. Затем команда запрашивает время жизни ключа:

Requested keysize is 2048 bits
Please specify how long the key should be valid.
0 = key does not expire
<n> = key expires in n days
<n>w = key expires in n weeks
<n>m = key expires in n months
<n>y = key expires in n years
Key is valid for? (0) 2m

В этом случае мы выбрали два месяца. Команда отвечает датой и временем через два месяца в будущем и запрашивает подтверждение.

Key expires at Sat 27 Apr 2015 11:14:17 AM BST
Is this correct? (y/N) y

На этом этапе команда **gpg** запрашивает информацию для идентификации ключа. Запрашиваемый здесь «ID пользователя» не связан с **UID** в стандартной базе данных аутентификации Linux. В этом примере ответы выделены жирным шрифтом:

Real name: Michael Jang
Email address: michael@example.com
Comment: DSA and Elgamal key
You selected this USER-ID:
"Michael Jang (DSA and Elgamal key) <michael@example.com>"
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O

Теперь система должна запросить пароль. Затем команда gpg начинает работать. Особенно с большими ключами может показаться, что они на несколько минут останавливаются с сообщением о создании случайных байтов. Возможно, вам придется запустить некоторые другие программы, чтобы стимулировать процесс. По завершении отобразится сообщение, подобное следующему:

gpg: key D385AFDD marked as ultimately trusted
public and secret key created and signed.

Чтобы убедиться, что открытый и закрытый ключи действительно записаны, выполните следующую команду:

$ gpg --list-key

Вывод должен включать самый последний ключ вместе с любыми другими, созданными из домашнего каталога пользователя, в подкаталоге **.gnupg/**. Для указанных параметров, если это единственная пара ключей в локальной учетной записи, вы увидите нечто похожее на следующий результат:

/home/michael/.gnupg/pubring.gpg
----
pub 2048D/9F688440 2015-04-28 [expires: 2015-06-27]
uid Michael Jang (DSA and ElGamal) <michael@example.com>
sub 2048g/306A91C0 2015-04-28 [expires: 2015-06-27]

Используйте ключ GPG2 для шифрования файла

Теперь вы можете отправить открытый ключ в удаленную систему. Чтобы начать процесс, вам нужно экспортировать открытый ключ. Для только что созданной пары ключей вы можете сделать это с помощью следующей команды (замените ваше имя на «Michael Jang»):

$ gpg --export Michael Jang > gpg.pub

Теперь скопируйте этот ключ в удаленную систему. Средство доставки, такое как электронная почта, флешка или показанная здесь команда **scp**, не имеет значения. Эта конкретная команда из учетной записи пользователя michael скопирует ключ **gpg.pub** в домашний каталог пользователя **michael** в системе **tester1.example.com**. Если вы предпочитаете, замените **IP-адрес**, например, так:

$ scp gpg.pub tester1.example.com:

Теперь перейдите к удаленной системе (в данном случае, **tester1.example.com**). Войдите в учетную запись пользователя **michael** (или домашний каталог учетной записи, в которую вы скопировали ключ **gpg.pub**). После подключения к этой системе сначала проверьте наличие ключей **GPG** с помощью следующей команды:

$ gpg --list-key

Если это система без предыдущих ключей **GPG**, список должен быть пустым, и ничего не будет отображаться в выходных данных этой команды. Теперь импортируйте файл gpg.pub в список локальных ключей **GPG** с помощью следующей команды:

$ gpg --import gpg.pub

Подтвердите импорт, снова выполнив команду **gpg --list-key**.
Теперь в удаленной системе вы можете зашифровать файл с помощью команды **gpg**. В следующем примере шифруется локальный файл **keepthitos.secret**:

$ gpg --out underthe.radar --recipient 'Michael Jang' --encrypt keepthis.secret

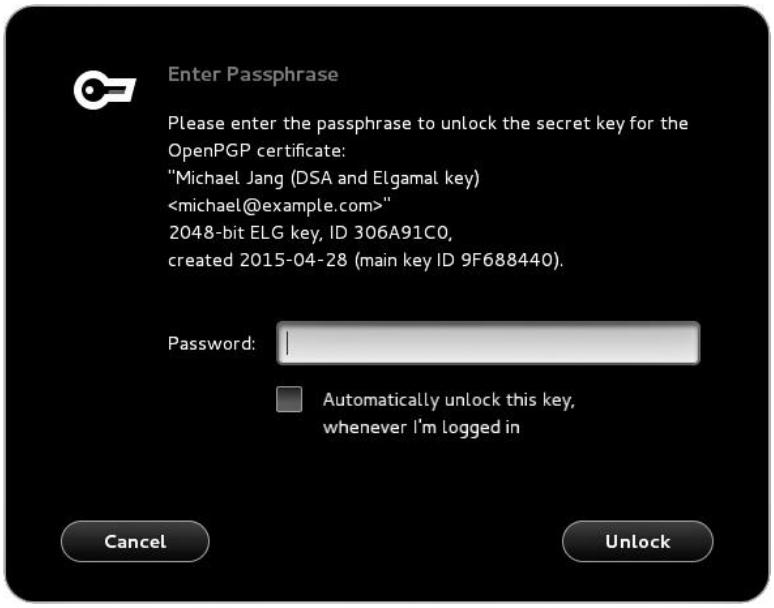
Имя пользователя в этом случае - Майкл Джанг. Если вы только что импортировали закрытый ключ, имя пользователя, как показано в выводе команды **gpg --list-key**, может отличаться. Заменить при необходимости.

Теперь, когда файл **underthe.radar** скопирован в исходную систему **server1.example.com**, вы можете запустить процесс расшифровки с помощью закрытого ключа с помощью следующей команды:

$ gpg --out keepthis.secret --decrypt underthe.radar

В консоли вам будет предложено ввести фразу-пароль, созданную ранее, с экраном, подобным показанному на **рисунке 10-14**.

РИСУНОК 10-14 Запрос парольной фразы для расшифровки



РЕЗЮМЕ СЕРТИФИКАЦИИ

Для защиты данных, сервисов и систем в сети Linux предоставляет уровни безопасности. Если служба не установлена, злоумышленник не может использовать ее для проникновения в систему. Те системы, которые установлены, должны быть обновлены. Такие услуги могут быть защищены брандмауэрами, а также с параметрами безопасности на уровне хоста и пользователя. Многие услуги включают в себя свои собственные уровни безопасности. RHEL 7 включает в себя несколько рекомендаций **АНБ**, в том числе **SELinux**.

Зональные межсетевые экраны могут регулировать и защищать шлюзы, а также отдельные системы. Этот же демон **firewalld** может использоваться для настройки пересылки пакетов, а также маскировка(**masquerading**) частных сетей. Такие параметры могут быть настроены непосредственно через **CLI** с помощью **firewall-cmd** или настроить с помощью инструмента настройки брандмауэра(**Firewall Configuration tool**).

Эти демоны, связанные с библиотекой **TCP Wrappers**, могут быть защищены соответствующими настройки в файлах **/etc/hosts.allow** и **/etc/hosts.deny**. Если есть конфликт между содержимым **/etc/hosts.allow** и **/etc/hosts.deny**, **/etc/hosts.allow** читается первым. Регулирование через **TCP Wrappers** возможно указывая пользователя или хост.

PAM поддерживает пользовательскую безопасность для ряда инструментов администрирования. Они настраивается индивидуально через файлы в каталоге **/etc/pam.d**. Эти файлы относятся к модулям в каталоге **/usr/lib64/security**.

Linux поддерживает шифрование с помощью **GPG. RHEL 7** включает в себя **GPG2** для этого цель, а также такие команды, как **gpg** для установки пар секретного/открытого(**private/public**) ключей с использованием шифрования RSA, DSA или схема Эль-Гамаля.

ПАРУ МИНУТ ПОВТОРЕНИЯ

Ниже приведены некоторые из ключевых моментов целей сертификации в главе 10.

Слои безопасности Linux

- Бастионные системы более безопасны, потому что они настроены на одну услугу. Благодаря виртуализации бастионные системы стали практичным вариантом даже для небольших
- организаций.
- Вы можете автоматизировать хотя бы обновления безопасности с помощью обновлений программного обеспечения Инструмент предпочтений.
- Многие сервисы включают свои собственные параметры безопасности в свои файлы конфигурации.
- Безопасность на основе хоста можно настроить по имени домена или IP-адресу.
- Безопасность пользователей включает в себя определенных пользователей и группы.
- **PolicyKit** может регулировать безопасность административных инструментов, запускаемых из **GNOME** окружение рабочего стола.

Межсетевые экраны и трансляция сетевых адресов

- Команда настройки **firewalld** - это **firewall-cmd**.
- С помощью **firewalld** вы можете назначать интерфейсы и исходные диапазоны IP-адресов различным зонам, как а также контролировать, какой трафик разрешен в зону.
- С помощью **firewalld** вы также можете маскировать IP-адреса из одной сети на
- вне сети, такой как Интернет.
- **firewalld** также можно настроить с помощью инструмента настройки брандмауэра, которую вы можете запустить с помощью команды **firewall-config**.

TCP Wrappers

- Команда **strings -f /usr/sbin/*** может идентифицировать службы, которые могут регулироваться **TCP Wrappers**.
- Клиенты и пользователи, перечисленные в **/etc/hosts.allow**, имеют доступ; перечисленные клиенты и пользователи в **/etc/hosts.deny** запрещен доступ.
- Не забудьте использовать фактическое имя исполняемого файла демона в **/etc/hosts.allow и /etc/hosts.deny** (обычно в **/usr/sbin**), например, **vsftpd**.

Сменные модули аутентификации

- RHEL 7 использует систему сменных модулей аутентификации (**Pluggable Authentication Modules PAM**) для обеспечения общий интерфейс прикладного программирования для служб аутентификации.
- Модули **PAM** вызываются файлами конфигурации в каталоге **/etc/pam.d**. Эти файлы конфигурации обычно называются как службы или команды, которыми они управляют.
- Существует четыре основных типа правил **PAM**: аутентификация, учетная запись, пароль и управление сессиями.
- Файлы конфигурации **PAM** включают в себя строки, в которых перечислены тип, **control_flag** и имя фактического модуля, за которым следуют необязательные аргументы.Модули
- **PAM** хорошо документированы в **/usr/share/doc/pam-versionnumber/txts** каталог и в справочных страницах.

Безопасные файлы и многое другое с GPG2

- **GPG** - это реализация **PGP** с открытым исходным кодом.
- **RHEL 7** включает в себя версию 2 **GPG**, известную как **GPG2**.
- Для шифрования и подписи **GPG2** могут использоваться ключи **DSA, RSA и ElGamal**.

- Ключи **GPG2** могут быть созданы с помощью команды **gpg --gen-key** и перечислены с помощью команды **gpg --list-key**.

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.

Слои безопасности Linux

1. Какой вариант безопасности лучше всего подходит для службы, которая в данный момент не требуется в системе?
   _____

Межсетевые экраны и трансляция сетевых адресов

2. Рассмотрим систему с настройками **firewalld** по умолчанию, в которой были использованы следующие команды:
   # firewall-cmd --zone =dmz --add-source=192.168.77.77
   # firewall-cmd --zone =dmz --remove-service=ssh
   После того, как они введены, и до перезагрузки, какой эффект они будут иметь, когда клиент с IP адрес 192.168.77.77 пытается подключиться к этой системе?
   _____

3. Какие каталоги содержат файлы конфигурации, которые определяют службы firewalld?
   _____

4. Вы открываете небольшой офис и хотите предоставить доступ в Интернет небольшому числу пользователи, но не имеют достаточно выделенных публичных адресов IPv4 для каждой системы в сети. Что ты можешь сделать?
   _____

5. Какой командный переключатель **firewall-cmd** настраивает маскировку?
   _____

6. Какая команда **firewall-cmd** добавляет расширенное правило в зону по умолчанию, чтобы разрешить HTTP-соединения только с хоста **192.168.122.50**?
   _____

7. Какой параметр команды **firewall-cmd** используется для внесения постоянных изменений в конфигурацию?
   _____

TCP Wrappers

8. С помощью файлов конфигурации **TCP Wrappers**, как вы можете ограничить **FTP-доступ** к клиентам на **192.168.170.0/24** сеть? Подсказка: **демон vsFTP**, если он установлен, находится в **/usr/sbin/vsftpd.**
   _____

9. Что происходит с сервисом, если вы разрешаете сервис в **/etc/hosts.allow** и запрещаете его в **/etc/hosts.deny**?
   _____

Сменные модули аутентификации

10. Каковы четыре основных типа правил **PAM**?
    _____
    _____
    _____
    _____

11. Вы редактируете файл конфигурации **PAM**, добавляя модуль. Какой контрольный флаг немедленно завершает процесс проверки подлинности, если модуль успешно?
    _____

Безопасные файлы и многое другое с GPG2

12. Какая команда перечисляет открытые ключи **GPG**, загруженные в текущую локальную учетную запись?
    _____

LAB ВОПРОСЫ

Некоторые из этих лабораторных работ включают упражнения, которые могут серьезно повлиять на систему. Вы должны сделать эти упражнения только на тестовых машинах. Вторая лаборатория главы 1 устанавливает KVM для этой цели.

Red Hat представляет свои экзамены в электронном виде. По этой причине лаборатории для этой главы доступны на носителе, сопровождающий книгу, в подкаталоге **Chapter10/**. Они доступны в **.doc, .html, и .txt** формате. Если вы еще не настроили **RHEL 7** в системе, обратитесь к первой лабораторной главе в главе 2. для инструкций по установке. Ответы для каждой лаборатории следуют за ответами Самотестирования для заполнения вопросы.

Labs

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

Лаборатория 1

В этой лабораторной работе вы настроите **пару ключей GPG** и протестируете результат в файле **19610-labs.txt**. Эта лаборатория требует использования двух разных систем. Системы **server1.example.com и tester1.example.com** идеально подходят для этой цели.

1. В системе **server1.example.com** скопируйте файл **19610-labs.txt** в домашний каталог обычного пользователя.
2. В системе **tester1.example.com** создайте пару пара закрытый-открытый ключ с использованием шифрования **RSA**.
3. Скопируйте открытый ключ в систему **server1.example.com**, а затем импортируйте его.
4. Зашифруйте файл **19610-labs.txt**.
5. Скопируйте зашифрованный файл в систему **tester1.example.com**.
6. Расшифруйте файл 19610-labs.txt в системе **tester1.example.com**.

Эта лабораторная проверка, конечно, тривиальна (она либо работает, либо нет), но «ответа» как такового нет - она требует возврата к тексту. Возможно, стоит ввести полные команды для генерации ключа, копирования ключа, его импорта, а затем для шифрования, копирования и дешифрования файла.

Лаборатория 2

Подумайте о бастионных системах. Один неявный урок этой главы - минимизировать то, что установлено в системе. Другими словами, если служба не установлена, хакер черной шляпы не может воспользоваться ею. В этой лабораторной работе используются команды **rpm**, описанные в главе 7. Хотя это тема **RHCSA**, они являются фундаментальными навыками для **RHCE**. Строго говоря, эта лаборатория не занимается непосредственно целями **RHCE**. Так что, если вам не хватает времени, пропустите эту лабораторию. Но это указывает на фундаментальный способ обеспечения безопасности системы **RHEL 7**.

Для этого просмотрите текущий вывод команды **systemctl list-unit-files --type = service**. Это удобный ориентир для установленных служб. Не удаляйте ничего в это время. Если вы не узнаете службу, определите связанный с ней пакет демона и RPM. Например, чтобы определить пакет, связанный со службой **abrtd**, выполните следующую команду:

$ rpm -qf /usr/lib/systemd/system/abrtd.service
abrt-2.1.11-12.el7.x86_64

Это идентифицирует пакет **abrt RPM**. Чтобы узнать больше об этом пакете, выполните следующую команду:

$ rpm -qi abrt

В рамках этой лабораторной работы вам не нужно повторять этот процесс для всех служб, установленных в системе. Например, посмотрите на следующие сервисы и получите больше информации о связанных RPM-пакетах, чтобы определить, могут ли они быть удалены из системы:

abrtd.service
atd.service
avahi-daemon.service
bluetooth.service

Лаборатория 3

Вы хотите настроить брандмауэр для размещения защищенного веб-сервера, который поддерживает входящие запросы как к обычному, так и к защищенному протоколам веб-сервера. Система также должна принимать удаленные соединения с локальным сервером SSH. Что вы делаете?

Лаборатория 4

Вы хотите настроить службу FTP во внутренней локальной сети, доступную только для одного определенного IP-адреса. Вы хотите заблокировать доступ из локальной сети. Предположим, что сетевой адрес вашей локальной сети - 192.168.122.0/24, а IP-адрес компьютера, который должен получить доступ, - 192.168.122.150. Для целей этой лабораторной работы вы можете заменить IP-адрес второго компьютера с Linux в локальной сети. Что вы делаете?

Лаборатория 5

Вы хотите использовать **TCP Wrappers** для ограничения доступа к локальному **SSH**-серверу. Во-первых, что вы делаете, чтобы подтвердить, что **SSH** может быть защищен **TCP Wrappers**? После этого внесите необходимые изменения, чтобы обеспечить доступ к серверу **SSH в системе server1.example.com** только из системы **tester1.example.com**.

Лаборатория 6

В этой лабораторной работе предполагается, что вы установили сервер **vsFTP**, как описано в главе 1. Основная задача этой лабораторной работы начинается с файла **/etc/vsftpd/ftpusers**. Как отмечено в этом файле, это список пользователей, которым не разрешено входить на локальный сервер **vsFTP**. В целях безопасности рекомендуется сохранить файл **/etc/vsftpd/ftpusers**.

Целью этой лабораторной работы является настройка и ограничение доступа через **FTP-сервер** к одной учетной записи обычного пользователя.

ОТВЕТЫ НА САМОПРОВЕРКУ

Слои безопасности Linux

1. Опция безопасности, которая лучше всего подходит для службы, которая в настоящее время не требуется в системе, заключается в установить этот сервис.

Межсетевые экраны и трансляция сетевых адресов

2. На основе данных команд любая попытка подключения к **службе SSH из 192.168.77.77** система отклонит.
3. Конфигурация служб **firewalld** хранится в **/usr/lib/firewalld/services/** и **/etc/firewalld/services/directory**.
4. Чтобы создать небольшой офис, предоставляя доступ в Интернет небольшому количеству пользователей, все, что вам нужно, это один выделенный IP-адрес. Другие адреса могут быть в частной сети. Маскарад делает это возможным.
5. Командный переключатель **firewall-cmd**, который устанавливает **маскарадинг**, **--add-\ masquerade.**
6. Следующее расширенное правило разрешает **HTTP-трафик с 192.168.122.50** для зоны по умолчанию:
   # firewall-cmd --add-rich-rule='rule family=ipv4 source address=192.168.122.50 service name=http accept'
7. Опция **--permanent firewall-cmd** используется для постоянной настройки изменения. Не забудьте загрузить сохраненную конфигурацию в конфигурацию во время выполнения, используя команду **firewall-cmd --reload**.

TCP Wrappers

8. Чтобы ограничить **FTP-доступ** для клиентов **в сети 192.168.170.0**, вы должны разрешить доступ к сети в **/etc/hosts.allow** и запретить это всем остальным в **/etc/hosts.deny**. Поскольку **/usr/sbin** находится в корневом пути пользователя, Вы можете напрямую ссылаться на демон **vsftpd** и добавить следующую директиву в **/etc/hosts.allow**:
   vsftpd: 192.168.170.0/255.255.255.0
   Затем вы добавили бы следующее в **/etc/hosts.deny**:
   vsftpd: ALL
9. Если вы разрешаете службу в **/etc/hosts.allow** и запрещаете ее в **/etc/hosts.deny**, служба разрешается.

Сменные модули аутентификации

10. Четырьмя основными типами правил **PAM** являются **auth, account, password и session**. Тип включения относится к одному или нескольким другим типам **PAM** в другом файле.
11. Флаг управления **sufficient** немедленно завершает процесс аутентификации, если модуль завершается успешно.

Безопасные файлы и многое другое с GPG2

12. Команда, которая перечисляет в настоящее время загруженные открытые ключи, является **gpg --list-key. Ключ gpg2 - list-key** команда также приемлема.

ОТВЕТЫ Лабораторной работы

Лаборатория 1

Проверка этой лаборатории должна быть простой. Если это работает, вы сможете подтвердить с помощью следующая команда в системе **tester1.example.com**:

$ gpg --list-keys

Он должен включать открытый ключ **GPG2**, только что импортированный в эту систему. Конечно, если шифрование, передача файлов и расшифровка работали, вы также должны иметь возможность читать расшифрованный текстовый файл в локальном Текстовом редакторе.

Лаборатория 2

Эта лаборатория говорит сама за себя, поскольку она может помочь вам подумать о том, как сделать систему более безопасный. Как уже говорилось в главе, вы можете начать с минимальной установки. Минимальная установка **RHEL 7** включает **SSH-сервер** для удаленного административного доступа.

Хотя **RHEL 7** значительно сократил количество установленных стандартных служб, большинство пользователей найдут некоторые услуги, которые не требуются. Например, сколько администраторов на самом деле нужно **Bluetooth** службы для системы **RHEL 7**, установленной на виртуальной машине?

Лаборатория 3

Если вы хотите настроить компьютер RHEL в качестве защищенного веб-сервера, это простой процесс, который описано в **главе 14**. Однако настройка брандмауэра является частью процесса, описанного в этой главе. Для этого вам нужно настроить брандмауэр для блокировки всех портов, кроме самых важных. Это должно включать Порты **TCP/IP 80 и 443**, которые позволяют внешним компьютерам получать доступ к сервису в локальной сети к простым и защищенным портам. Открытые порты также должны включать порт 22 для связи SSH.

Самый простой способ настроить это с помощью инструмента настройки **Red Hat Firewall,** который вы можете запустить с помощью команды **firewall-config**. В средстве настройки брандмауэра выполните следующие действия, которые немного различаются в графической и консольной версиях инструмента:

1. Установите режим конфигурации на постоянный(**Permanent**).
2. Выберите «публичную **public**» зону. Перейдите на вкладку <**interface**> и убедитесь, что сетевые интерфейсы системы перечислены в этой зоне.
3. Выберите вкладку «Службы(**Services**)» и включите службу **http**. Это позволяет получить доступ извне компьютер на местном регулярном сайте. Активируйте опцию **https**. Убедитесь, что опция **ssh** остается активным.
4. Нажмите Параметры(**Options**) | Перезагрузите (Reload) Firewalld, чтобы применить изменения к конфигурации времени выполнения.
5. Введите следующую команду, чтобы проверить полученный брандмауэр:
   # firewall-cmd --list-all
6. После настройки веб-службы, как описано в **главе 14**, пользователи смогут получить доступ к как обычные, так и защищенные веб-серверы от удаленных систем.

Лаборатория 4

Следующие шаги демонстрируют два разных способа ограничения доступа к отмеченной системе по **IP-адресу192.168.122.150**. Любой из двух методов будет приемлемым. Эти методы защищают **vsFTP** в двух способы: через **TCP Wrappers** и с помощью соответствующих команд брандмауэра. В сценарии «**real-world**»

Вы можете использовать все методы в многоуровневой стратегии безопасности. Эти шаги предполагают, что вы выполняете эту лабораторию в системе server1.example.com.

1. Убедитесь, что **RPM vsftpd** установлен.
2. Запустите службу **FTP**. Используйте команду **systemctl start vsftpd**.
3. Сделайте резервную копию текущей версии файла **/etc/hosts.deny**. Откройте этот файл в текстовом редакторе. Добавить **vsftpd : ALL EXCEPT 192.168.122.150**.
4. Попробуйте получить доступ к службе **FTP** с компьютера с **IP-адресом 192.168.122.50**. Что происходит? Попробуйте еще раз с другого компьютера в вашей локальной сети.
5. Восстановите предыдущий файл **/etc/hosts.deny**.
6. Заблокируйте службу **FTP** для всех **IP-адресов, кроме 192.168.122.150**, с помощью следующих команд:
   # firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=192.168.122.150 service \ name=ftp drop'
   # firewall-cmd --permanent --add-service=ftp
7. Активизируйте постоянную конфигурацию в конфигурацию времени выполнения с помощью команда **firewall-cmd --reload**.
A. Попробуйте получить доступ к **FTP-серверу** с компьютера с **IP-адресом 192.168.122.150**. Что происходит? Попробуйте еще раз с другого компьютера в локальной сети.
8. Удалите добавленные вами правила брандмауэра.
9. Бонус: повторите эти команды для службы **SSH на порту 22**.

Лабораторная 5

Чтобы убедиться, что **TCP Wrappers** можно использовать для защиты службы **SSH**, выполните следующею команду:

 # ldd /usr/sbin/sshd | grep libwrap

Вывод, который включает ссылку на библиотеку **libwrap.so.0**, подтверждает ссылку на **TCP Wrappers**
библиотека. В целом, безопаснее всего запретить доступ ко всем службам, включив следующую запись в
Файл **/etc/hosts.deny**:

ALL:ALL

Затем вы можете настроить доступ к службе **SSH** с помощью следующей строки в файле **/etc/hosts.allow**:

sshd: 192.168.122.50

При этом в большинстве случаев используется полное доменное имя для указанного IP-адреса (**server1.example.com**) тоже должно работать, использование IP-адреса часто уместно. Ограничения по IP-адресу не зависит от того, насколько точны соединения с **DNS**-серверами или обратный **DNS**.

Конечно, это не единственный способ ограничить доступ к службе SSH для одной системы. Возможно в файле **/etc/hosts.deny** с такой директивой:

sshd: ALL, EXPECT 192.168.122.50

Это можно настроить с помощью других параметров безопасности, таких как служба на основе зоны межсетевого экрана.

Лаборатория 6

Прежде чем эта лаборатория сможет работать, вам необходимо активировать один логический элемент **SELinux: ftp_home_dir**. Это перечислено в инструменте администрирования **SELinux** как «Определите, может ли ftpd читать и записывать файлы в домашней папке пользователя». каталоги». Поэтому, с идентифицированным логическим ключом, вы сможете настроить **vsFTP**, как описано.

Описание в этой лабораторной работе должно указывать на файл конфигурации **/etc/pam.d/vsftpd**. Модель командная строка в этом файле требуется авторизация

auth required pam_listfile.so item=user sense=deny file=/etc/vsftpd/ftpusers onerr=succeed

который указывает на файл /**etc/vsftpd/ftpusers**, список пользователей, которым «**отказано**» в доступе. Как условия в Лабораторной работе предполагает, что вам нужен список (одного) пользователя, доступ к которому должен быть разрешен, вторая строка аналогична введите в этом файле соответствующий. Например,

auth required pam_listfile.so item=user sense=allow file=/etc/vsftpd/testusers onerr=succeed

позволяет всем пользователям, перечисленным в файле **/etc/sftpd /testusers**. Директива **onerr=success** означает, что Сервер **vsFTP** все еще работает, если есть ошибка в другом месте. Например, если нет файла **testusers** в Каталоге **/etc/vsftpd**, директивы в этой строке прощают, что позволяет условие **auth** модуля.

В качестве эксперимента попробуйте эту лабораторную работу с булевой переменной **ftp_home_dir**, установленной и неустановленной. Это должно продемонстрировать силу **SELinux** и послужить соответствующим предварительным просмотром **главы 11**.