СОДЕРЖАНИЕ

1 Аннотация

В данном документе приведено описание средств ROSA Enterprise Linux Server для обеспечения защиты данных и разграничения доступа. В руководстве приведено краткое описание простейших средств обеспечения безопасности, отдельно рассмотрены принципы использования SELinux. Руководство предназначено для пользователей, знакомых с базовыми возможностями ОС Linux.

1.1 Выделение важной информации

В документе для выделения информации, на которую стоит обратить внимание, используются примечания и иконки  ,  ,  .

Примечания выделяются подчеркиванием текста и содержат дополнительную информацию о конфигурировании RELS или о дополнительных возможностях команд.

Пример:

> Примечание: на nfs-клиенте просмотреть расшаренные папки можно командой
>
> `showmount –e 192.168.68.128`
>
> где `192.168.68.128` — адрес nfs-сервера.

Иконка  служит для выделения возможностей RELS, которые существенно упрощают работу с операционной системой, или сведений о готовой конфигурации, которую можно использовать при работе с RELS.

Пример:

>  Пример содержимого конфигурационного файла для кластера на двух хостах:
>
> ```
> <cluster name="mycluster" config_version="2">
>    <cman two_node="1" expected_votes="1"/>
>    <clusternodes>
>      <clusternode name="rosa2.int" nodeid="1">
>          <fence>
>          </fence>
>      </clusternode>
>      <clusternode name="rosa.int" nodeid="2">
>          <fence>
>          </fence>
>      </clusternode>
> ```

```
    </clusternodes>
    <fencedevices>
    </fencedevices>
    <rm>
    </rm>
</cluster>
```

Иконка  служит для того, чтобы обратить ваше внимание на те или иные особенности работы RELS. Эта информация не является критически необходимой, но мы рекомендуем строго следовать советам, приведенным с этой иконкой. Это поможет сэкономить время.

Пример:

>  Перед запуском в промышленную эксплуатацию GFS мы рекомендуем проконсультироваться со специалистами технической поддержки РОСЫ, а также провести опытную эксплуатацию инфраструктуры с GFS в течении 2-3-х месяцев, чтобы понаблюдать за устойчивостью работы хранилища.

Иконка  служит для выделения критически важной информации. Внимательно читайте эту информацию и строго следуйте рекомендациям. В противном случае у вас могут возникнуть серьезные сбои или просто не запустятся важные сервисы RELS.

Пример:

>  Примечание: иногда при соединении с хостом кластера через интерфейс luci возникает ошибка “`authentication to ricci agent failed`”. Данная проблема решается разрешением доступа к нужному порту в файерволе, а также установкой пароля ricci командой:
>
> ```
> passwd ricci
> ```
>
> на каждом хосте. Иногда также в этой ситуации стоит отключить selinux, если вы не уверены, что абсолютно правильно пользуетесь им.

1.2 Ссылки на другие разделы документа и рисунки

В документе используются ссылки на рисунки и другие разделы документа, для перехода по ссылке в PDF-версиях документа необходимо нажать клавишу CTRL и щелкнуть левой кнопкой мыши на ссылку.

Желаем приятного знакомства с возможностями RELS!

2 Основы безопасности

2.1 Введение

Для обеспечения информационной безопасности при эксплуатации RELS рекомендуется соблюдать хотя бы следующий набор минимальных правил:

– строго следовать политикам безопасности при разграничении доступа пользователей к информации;

– контролировать цифровые подписи устанавливаемого и обновляемого ПО;

– осуществлять защиту интерфейсов операционной системы путем настройки брандмауэров, создавать защищенную конфигурацию соответствующих служб;

– обеспечить защиту внешнего контура операционной системы (защита загрузчика);

– устанавливать серверные службы совместно с антивирусом.

2.2 Аутентификация и идентификация

Механизмы идентификации и аутентификации являются обязательными компонентами модели защиты (и, соответственно, RELS). Ни один пользователь не может начать работу с операционной системой, не идентифицировав себя и не предоставив информацию аутентификации, подтверждающую, что пользователь действительно является тем, за кого себя выдает.

Каждый пользователь имеет свой уникальный числовой идентификатор (UID) — натуральное число, которое обычно выбирается автоматически при регистрации учетной записи. Это число не может быть произвольным, поскольку в RELS существуют правила, определяющие, каким типам пользователей могут быть выданы идентификаторы из того или иного диапазона.

Идентификатору пользователя соответствует системное имя пользователя (учетная запись). Для привилегированного пользователя с учетной записью root зарезервирован нулевой идентификатор.

Для более удобного управления доступом к ресурсам все пользователи включаются в группы. Группа — это подмножество пользователей, объединенных по каким-либо критериям. У группы так же, как и у пользователя, есть имя и идентификационный номер — GID. В операционной системе каждый пользователь должен принадлежать как минимум к одной группе — группе по умолчанию. При создании учетной записи пользователя обычно создается и группа, имя которой совпадает с системным именем пользователя. Именно эта группа будет использоваться как группа по умолчанию для данного пользователя. Макси-

мальное количество групп, в которых может состоять один пользователь, равно 32.

Аутентификация — это процесс установления подлинности пользователя. Для аутентификации в RELS используется пароль. Пароль представляет собой набор символов, известный только его владельцу и используемый для удостоверения его подлинности. Каждый пользователь имеет собственный пароль. Наличие пароля — необходимая составляющая политики безопасности пользователей. Без пароля, зная только имя пользователя, осуществить вход невозможно.

Требования к паролю:

- **секретность**;
- **устойчивость к угадыванию**;
- **устойчивость к атаке перебором**.

2.3 Управление учетными записями пользователей

Все имена пользователей и соответствующие им идентификаторы хранятся в файле `/etc/passwd`. Пример содержимого файла:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/sh
daemon:x:2:2:daemon:/sbin:/bin/sh
adm:x:3:4:adm:/var/adm:/bin/sh
<...>
messagebus:x:499:499:system user for dbus:/:/sbin/nologin
rpm:x:498:498:system user for rpm:/var/lib/rpm:/bin/false
<...>
live:x:500:500::/home/live:/bin/bash
user:x:501:501:User:/home/user:/bin/bash
```

Каждая запись в этом файле состоит из 7 полей, разделенных символом «:».

1) Системное имя пользователя.

2) Поле пароля. В ранних версиях данное поле содержало зашифрованный пароль, но после введения технологии теневых паролей в нем ставится символ «x».

3) Поле идентификатора пользователя (UID). Каждый пользователь имеет свой уникальный идентификационный номер. Этот номер используется в различных целях, например, при установке прав доступа на файлы.

4) Поле идентификатора группы (GID). В этом поле указывается группа, к которой принадлежит пользователь.

5) Поле комментария. Используется (опционально) для хранения дополнительной информации о пользователе, например, его полного имени.

6) Поле пути к домашнему каталогу.

7) Поле пути к командной оболочке. Содержит полный путь к рабочей оболочке пользователя (по умолчанию такой оболочкой является bash). Эта оболочка запускается, когда пользователь проходит процедуру аутентификации. В целях безопасности для системных пользователей в этом поле рекомендуется указать `/sbin/nologin`. Сама по себе программа **nologin** не является оболочкой, единственное ее назначение — не допустить вход в ОС. При попытке входа под именем пользователя, у которого в качестве рабочей оболочки установлена `/sbin/nologin`, ничего не происходит. Также в данное поле можно установить значение `/bin/false`. В RELS успешное завершение программы определяется типом возвращаемого значения. Если возвращается нулевое значение, это означает, что выполнение программы прошло успешно. Если ненулевое — значит, в процессе выполнения программы произошли ошибки. На основе возвращаемого значения система аутентификации делает вывод о том, пройдена аутентификация успешно или нет. Программа **false** независимо от внешних факторов возвращает значение, отличное от нуля, что означает возникновение ошибок при запуске оболочки и возврат управления системе аутентификации.

При авторизации производится чтение информации о пользователях из файла `passwd`. Право на запись в этот файл имеет только пользователь `root`; читать этот файл могут все пользователи.

Для управления пользователями и группами используются утилиты `useradd`, `usermod`, `userdel`, `getent`, `groupadd`, `groupdel`, `groupmod`.

Пароли хранятся в специальном файле `/etc/shadow` в неявном виде, что обеспечивает их защиту. Для генерации файла `/etc/shadow` используется утилита `pwconv`, а для отказа от использования этого файла — `pwunconv`.

Файл `shadow`, как и файл `passwd`, разделен на несколько полей символом «:».

1) Имя пользователя. Это поле просто дублируется из файла passwd.

2) Хеш пароля. Пароль, в отличие от имени пользователя, никогда не хранится в открытом виде. При установке пароля до сохранения его в файле он шифруется по специальному алгоритму. По умолчанию таким алгоритмом является алгоритм одностороннего шифрования DES (Data Encryption Standard). Использование одностороннего алгоритма шифрования исключает возможность расшифровки пароля. Если данное поле содержит знак `!` или `*`, это означает, что учетная запись заблокирована и пользователь не сможет осуществить вход. Если поле содержит `!!`, это означает, что у пользователя никогда не было пароля и, не назначив его, он не сможет осуществить вход.

3) Дата последней смены пароля. В этом поле записывается число дней, прошедших

с 1 января 1970 г. до даты, когда пользователь сменил пароль в последний раз. Эта информация используется вместе со следующими полями, управляющими сроком действия пароля.

4) Число дней, которое должно пройти до смены пароля. Минимальный срок (в днях), который должен истечь, прежде чем пользователь сможет сменить пароль.

5) Число дней, после которого необходимо сменить пароль. Максимальный срок (в днях), по истечении которого необходимо сменить пароль.

6) Число дней до предупреждения о необходимости смены пароля. Число дней до истечения срока действия пароля, в течение которых пользователь будет получать предупреждения о скором окончании срока действия пароля.

7) Число дней до отключения учетной записи. Число дней, которое должно пройти с момента окончания срока действия пароля до отключения учетной записи.

8) Дата блокировки учетной записи. Дата (указанная в днях, прошедших с 1 января 1970 г.), когда учетная запись пользователя будет (или была) отключена.

9) Зарезервированное поле. Это поле игнорируется.

Пример строки файла `/etc/shadow`:

```
tester:$1$.QKDPc5E$SWlkjRWexrXYgc98F.:15631:0:60:7:10:15695:
```

– пароль был в последний раз изменен 19 октября 2012 г.;

– срок, в течение которого нельзя изменить пароль, не определен;

– пароль должен меняться каждые 60 дней;

– пользователь будет получать предупреждение о необходимости его сменить в течение 7 дней;

– учетная запись будет отключена через 10 дней после истечения срока действия пароля, если не будет попыток входа;

– срок действия учетной записи истекает 21 декабря 2012 г.

Для изменения временных параметров учетной записи пользователя используется утилита `chage`. Для получения руководства по использования утилит командной строки необходимо выполнить следующую команду:

```
man <имя_утилиты>
```

Файл паролей имеет права только на чтение и только для пользователя `root`. Для изменения пароля используется специальная утилита `passwd`, которая не дает установить легко взламываемый пароль. В качестве параметра в командной строке она получает имя

пользователя и при запуске требует ввода пароля для этого пользователя. В целях безопасности пароль при вводе не отображается на экране, поэтому существует очень высокая вероятность допустить ошибку, особенно если пароль состоит из цифр и символов в различном регистре. Поэтому ввод пароля для надежности осуществляется дважды. После подтверждения пароль шифруется и сохраняется в файле `/etc/shadow`.

В RELS имеется возможность ограничивать срок действия пароля (т. е. указывать время, в течение которого пароль считается верным). В конце срока действия пароля пользователю будет предложено ввести новый пароль, который затем может быть использоваться, пока срок его действия также не закончится. При выборе срока действия пароля следует соблюдать разумный баланс между безопасностью и удобством для пользователей. Также необходимо вести историю ранее использованных паролей, чтобы избежать их повторения в будущем.

При попытке входа набранный пароль снова шифруется и сравнивается с записью в файле, хранящем хеши паролей. Совпадение означает, что пароль введен верно, и доступ к RELS разрешается.

Для проверки синтаксиса файлов паролей используются утилиты `pwck` и `grpck`.

Утилита pwck последовательно анализирует записи файлов `/etc/passwd` и `/etc/shadow`, проверяя, что каждая запись содержит:

- правильное количество полей;
- уникальное имя пользователя;
- действительные идентификаторы пользователей и групп;
- действительную первичную группу;
- действительный домашний каталог;
- действительный командный процессор.

Утилита `grpck` выполняет проверку файлов `/etc/group` и `/etc/gshadow`. Она последовательно анализирует записи файлов и проверяет, что каждая запись содержит:

- правильное количество полей;
- уникальное имя группы;
- действительный список членов и администраторов группы.

2.4 Управление учетными записями пользователей в графическом интерфейсе

В RELS предусмотрена утилита для управления учетными записями пользователей и группами в графическом интерфейсе. Утилита называется **Менеджер пользователей**, для ее запуска необходимо выбрать в главном меню RELS пункты: «Администрирование»

- «Пользователи и группы». Чтобы получить доступ к утилите необходимо ввести пароль root. Интерфейс утилиты представлен на рисунке ниже.

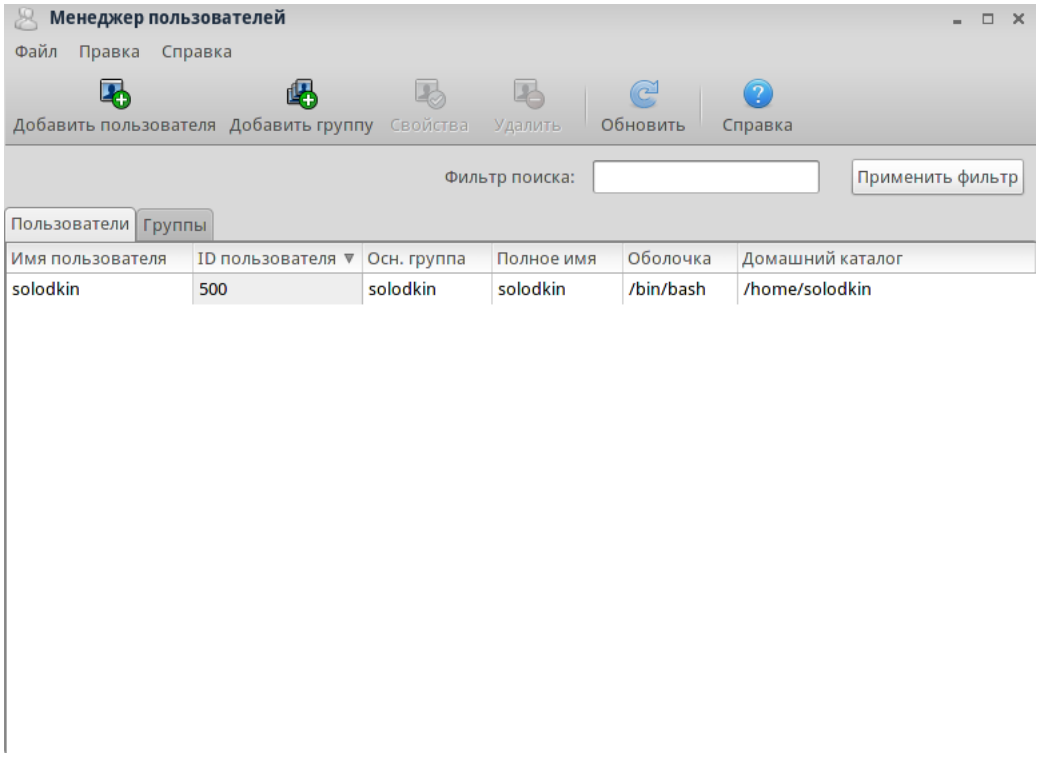

Рисунок 1

В принципе, данная утилита позволяет производить все необходимые базовые операции по управлению пользователями и группами, такие как (но не ограничиваясь перечисленными):

- Создание/удаление учетных записей;
- Редактирование учетных записей;
- Установка ограничений на время использования паролей;
- Создание/удаление/управление группами.

Чтобы отредактировать существующую учетную запись пользователя, необходимо дважды нажать на нее мышью, в открывшемся окне (Рисунок 2 Свойства пользователя), вы можете изменить имя пользователя и его пароль, отредактировать домашний каталог и используемую оболочку. Во вкладке «Сведения об учетной записи» можно ограничить срок существования учетной записи. Во вкладке «Сведения о пароле» — ограничить срок действия пароля. Во вкладке «Группы» можно изменить членство пользователя в группах.

Несмотря на то, что данная утилита предоставляет все необходимые минимальные функции по управлению пользователями и группами, мы рекомендуем использовать для этих целей значительно более широкий функционал MMC (см. Руководство пользователя RELS).

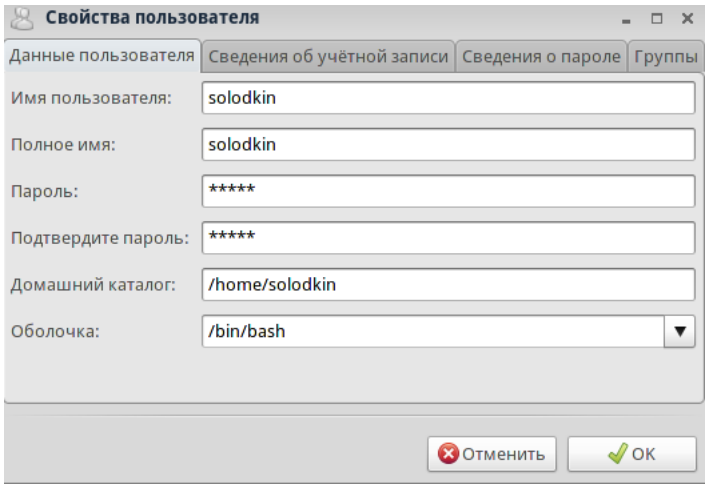

Рисунок 2

2.5 Права доступа к объектам ФС

Концепция политики файловой безопасности RELS строится на том, что любой файл предусматривает три группы правил доступа: для владельца файла (или его создателя), для группы пользователей (в которую чаще всего входит и владелец файла) и для всех остальных.

Права доступа к файлу описываются тремя восьмеричными цифрами. Первая — права доступа владельца, вторая — права доступа группы, третья — права доступа всех остальных. Каждая из этих восьмеричных цифр представляет собой маску из трех бит. Эти биты отвечают за права на чтение, на запись и на исполнение файла. Если бит равен 1, операция разрешена, если 0 — запрещена.

Очень часто вместо восьмеричных цифр приводятся строки из девяти символов `r`, `w`, `x` или `-`. Их можно разбить на три группы по 3 символа по тому же принципу, что и восьмеричные цифры. В этом случае внутри каждой группы на первой позиции будет стоять символ, отвечающий за чтение (r), на второй — за запись (w), а на третьей — за исполнение (x). При отсутствии соответствующего разрешения вместо буквы будет стоять прочерк.

Для файлов и каталогов (представляющих собой особый вид файла) влияние бит доступа несколько различаются.

Право на чтение файла позволяет пользователю видеть содержимое файла. Для каталога установка права на чтение означает, что разрешается читать файлы, находящиеся в этом каталоге.

Право на запись файла позволяет пользователю изменять его содержимое. Для каталога — создавать файлы внутри каталога.

Право на выполнение файла позволяет запускать его в качестве исполняемого. Для каталога установка этого права дает возможность входить в каталог и просматривать его

содержимое.

Предоставление прав доступа к файлу также зависит от прав доступа к каталогу, в котором находится файл. Например, даже если права доступа к файлу выглядят как `rwxrwxrwx` (всем все разрешено), другие пользователи не смогут получить доступ к файлу, пока у них не будет прав на чтение и исполнение применительно к каталогу, в котором находится этот файл.

Для того чтобы получить доступ к файлу, нужно иметь права на исполнение для всех каталогов на пути к нему и право на чтение (или исполнение) самого файла.

Обычно права доступа к пользовательским файлам имеют вид `rw-r-r-`, что позволяет другим пользователям читать эти файлы, но не изменять их. Для каталогов права доступа обычно задаются как `rwxr-xr-x`, что позволяет другим пользователям просматривать их, но не создавать и не удалять файлы в них.

Чтобы исключить возможность доступа других пользователей к тем или иным файлам, следует задать для них права доступа в виде `rw---`. Аналогичным образом установка прав доступа к каталогу в виде `rwx---` защитит этот каталог от вторжения извне.

Для просмотра прав доступа к файлу необходимо выполнить команду `ls -l`.

Изменение прав доступа к файлам осуществляется при помощи команды `chmod`. Права доступа к файлу при вызове команды задаются одним из двух способов: или битовой маской в десятичном представлении, или при помощи символов. Права доступа к файлу может изменять только владелец файла или администратор.

Синтаксис команды:

```
chmod [who] {+|-|=} [perm] <имя_файла_1> ... <имя_файла_N>
```

Параметр `[who]` определяет, для каких категорий пользователей устанавливаются права доступа. Он представляет собой один или несколько следующих символов:

a	—	установка прав доступа для всех категорий пользователей. Если параметр who не задан, по умолчанию он заменяется на a.
u	—	установка прав доступа для владельца файла
g	—	установка прав доступа для пользователей, входящих в группу владельца файла
o	—	установка прав доступа для всех остальных пользователей

Операция, выполняемая над правами доступа для заданной категории пользователей, определяется одним из следующих символов:

+	—	добавление прав доступа
-	—	отмена прав доступа
=	—	отмена всех существующих прав доступа и добавление перечисленных. Если параметр perm не определен, все существующие права доступа отменяются

Параметр [perm] определяет права доступа, которые будут добавлены, отменены или установлены взамен существующих, и представляет собой комбинацию одного или нескольких существующих символов:

r	—	право на чтение
w	—	право на модификацию
x	—	право на исполнение

Параметры <имя_файла_1> ... <имя_файла_N> — имена файлов, для которых производится изменение прав доступа. Вместо имен файлов могут использоваться шаблоны.

Для решения задач, связанных с правами доступа, также применяются команды chown (изменение владельца файла) и chgrp (изменение группы, которой принадлежит файл).

Синтаксис команды chown:

```
chown owner <имя_файла_1> ... <имя_файла_N>
```

Параметр owner задает нового владельца файла в символьном или в числовом виде (ID). Изменить владельца может только владелец файла или администратор. Вместо имен файлов могут использоваться шаблоны.

Синтаксис команды chgrp:

```
chgrp group <имя_файла_1> ... <имя_файла_N>
```

Команда chgrp устанавливает индикатор группы для указанных файлов равным индикатору группы, переданной в качестве параметра group. В качестве параметра group допускается использовать имя группы или ее числовой идентификатор.

Для получения руководства по использования утилит командной строки необходимо выполнить следующую команду:

```
man <имя_утилиты>
```

Команда umask применяется для определения режима по умолчанию для создания файлов, или маски прав доступа. Стандартные настройки umask задаются в файле /etc/profile и применяются ко всем пользователям RELS.

Чтобы рассчитать маску прав доступа, нужно поразрядно вычесть требуемую значение прав доступа файла из восьмеричного числа 777. Если маска равна 777, осуществить

какие-либо действия с файлами будет невозможно (777 – 777 = 000). С маской 666 вновь созданные файлы будут иметь маску 111.

Помимо прав доступа к файлу существуют так называемые модификаторы доступа. К модификаторам доступа относятся Sticky Bit, SUID и SGID.

Установка Sticky Bit для каталога позволяет пользователю записывать файлы в этот каталог, но удалять из него он сможет только те файлы, для которых он является владельцем или имеет явно заданные права записи.

Если для исполняемого файла установлен модификатор доступа SUID, он будет запускаться не с правами вызвавшего его пользователя, а с правами владельца файла. Такой прием используется для того, чтобы рядовой пользователь мог работать с некоторыми системными файлами, владельцем которых является `root`. Например, чтобы пользователь мог самостоятельно изменить свой пароль при помощи программы `passwd`, у этой программы, владельцем которой является пользователь `root`, должен быть установлен бит SUID, поскольку она работает с файлом `shadow`, модификацию которого имеет право производить только пользователь `root`.

По схожему принципу RELS действует и при установке модификатора доступа SGID, только в этом случае вместо владельца файла используется группа, к которой принадлежит файл. Если SGID установлен для каталога, файлы, содержащиеся в этом каталоге, будут иметь те же установки группы, что и каталог.

2.6 Защита загрузчика

Загрузчик ОС Linux обычно защищается паролем для достижения следующих целей:

1) *Предотвращение доступа к монопольному режиму*. Если взломщику удастся загрузить систему в монопольном режиме, он автоматически войдет в систему под именем `root`, не вводя пароля `root`.

2) *Предотвращение доступа к консоли GRUB*. Если в качестве загрузчика на компьютере используется GRUB, взломщик может воспользоваться редактором GRUB для изменения его настроек или сбора информации с помощью команды `cat`.

3) *Предотвращение доступа к небезопасным ОС*. Если на компьютере установлено несколько систем, взломщик сможет выбрать при загрузке другую систему, например, DOS, в которой игнорируются права доступа и разрешения для файлов.

Для защиты загрузчика GRUB необходимо задать пароль в его файле конфигурации (`/boot/grub/grub.conf`). Для этого в консоли под именем суперпользователя `root` ввести:

`/sbin/grub-md5-crypt`

Получив приглашение, ввести пароль GRUB и нажать `[Enter]`. В ответ будет выведен MD5-хэш пароля.

Затем отредактируйте файл конфигурации GRUB `/boot/grub/grub.conf`. Откройте файл и найдите в основном разделе ниже строки `timeout` вставьте строку `password --md5 <password-hash>`.

Заменить `<password-hash>` значением, возвращенным командой `/sbin/grub-md5-crypt`.

При следующей загрузке системы меню GRUB запретит вызов редактора или командной строки. Для получения доступа необходимо к редактору и командной строке необходимо нажать сначала `[p]` и затем ввести пароль GRUB.

Кроме того, необходимо изменить другую часть файла `/boot/grub/grub.conf`.

Необходимо найти строку `title` небезопасной ОС и добавить прямо под ней строку `lock`.

Чтобы задать для конкретного ядра или другой системы отдельный пароль, после строки `lock` необходимо добавить строку с паролем.

2.7 Рекомендации по созданию паролей

Пароли — это основной способ определения подлинности пользователя, поэтому парольная защита чрезвычайно важна для защиты пользователя, рабочей станции и сети.

Для создания безопасного пароля рекомендуется следовать следующим правилам:

1) Придумывать пароль длиной не меньше восьми символов.
2) Смешивать буквы верхнего и нижнего регистра (Av3TdYen1).
3) Смешивать буквы и цифры.
4) Включать не алфавитно-цифровые символы
5) Придумать пароль, который возможно запомнить.
6) Не записывать пароль на бумаге.
7) Не использовать словарные и общеизвестные слова в пароле

В случае если при создании нового пароля пользователя RELS идентифицирует, что пароль является небезопасным, то в консоли или терминале появится соответствующее предупреждение, в этом случае рекомендуется использовать другой пароль.

Кроме этого, рекомендуется использовать ограничение срока действия пароля. Ограничение срока действия означает, что по истечении заданного времени (обычно 90 дней) пользователю предлагается сменить пароль. При этом, если пользователь будет вынуж-

ден периодически менять пароль, подобранный взломщиком, то пароль будет действовать ограниченное время.

Для определения срока действия пароля используются команда:

`chage`

Параметр "`-M`" команды chage определяет максимальный срок действия пароля в днях. Например, чтобы срок действия пароля пользователя истекал через 90 дней, следует использовать команду:

```
chage -M 90 <username>
```

2.8 Ограничение доступа к правам root

Ограничение доступа к права суперпользователя (root) может существенно повысить безопасность эксплуатации ОС.

В таблице указаны способы защиты от входа под именем `root`.

Таблица 1

Способ	Необходимые действия	Что обеспечивает
Изменение оболочки root	Отредактировать файл /etc/passwd и сменить оболочку с /bin/bash на /sbin/nologin.	Предотвращает вход в оболочку root и регистрирует попытки входа. Программы `login`, `gdm`, `kdm`, `xdm`, `su`, `ssh`, `scp`, `sftp` не смогут работать с учетной записью `root` Примечание: в RELS по умолчанию используется графическая оболочка LXDE
Запрет доступа `root` с любых консольных устройств (tty)	Пустой файл `/etc/securetty` не позволит `root` войти в систему с любых устройств, подключенных к компьютеру	Предотвращает вход под именем `root` с консоли или по сети. Программы `login`, `gdm`, `kdm`, `xdm` не смогут работать с учетной записью `root`. Другие сетевые службы, открывающие `tty`.
Запрет входа root через SSH	Отредактировать файл `/etc/ssh/sshd_config` и задать параметр `PermitRootLogin` равным `no`	Предотвращает доступ `root` с помощью набора инструментов OpenSSH. Программы `ssh`, `scp`, `sftp` не смогут работать с учетной записью `root`

Дополнительно рекомендуется ограничить доступ к правам `root`, сделав его возмож-

ным только для определенной группы пользователей.

Это можно сделать, добавив пользователей в специальную административную группу `wheel`. Для этого необходимо выполнить от имени `root` следующую команду:

```
usermod –G wheel <username>
```

2.9 Защита оболочек TCP

Оболочки TCP способны не только ограничивать доступ к службам, но и могут использоваться для выдачи баннеров соединений, предупреждать об атаках с определенных узлов и осуществлять улучшенное ведения журнала.

Чтобы сделать для службы баннер с помощью TCP оболочек, необходимо воспользоваться параметром `banner`. Кроме того, необходимо создать специальный файл, который может находиться где угодно в системе, но должен носить имя «демона», например, `/etc/banners/vsftpd`. Пример содержимого такого файла:

```
220Hello, %c
220All activity on ftp.example.com is logged.
220Act up and you will be banned.
```

Вместо маркера `%c` будут подставлены сведения о клиенте, например, имя пользователя и компьютера или его IP-адрес, что делает соединение еще более опасным.

Чтобы этот баннер выдавался для всех входящих соединений, добавьте в файл `/etc/hosts.allow` следующую строку:

```
vsftpd : ALL : banners /etc/banners/
```

Если при попытке атаки на сервер были идентифицированы узлы или сети, оболочки TCP, следуя указанию `spawn`, могут сообщить о последующих атаках этих узлов или сетей.

Предположим, что при попытке нападения на сервер был идентифицирован взломщик из сети `206.182.68.0/24`. Следующая строка, помещенная в файл `/etc/hosts.deny`, запрещает подключение и регистрирует попытку в специальном файле:

```
ALL : 206.182.68.0 : spawn /bin/ 'date' %c %d >> /var/log/intruder_alert
```

Вместо маркера `%d` необходимо подставить имя службы, к которой пытался обратиться взломщик.

Чтобы разрешить подключение с регистрацией в журнале, необходимо поместить указание spawn в файл `/etc/hosts.allow`.

Для улучшенного управления доступом рекомендуется так же использовать службы `xinetd`.

Xinetd позволяет ограничить доступ с определенных узлов к службам под управлением xinetd, что, в свою очередь, ограничивает возможность сканирования портов сервера. Для того чтобы запретить доступ с этих узлов к службам xinetd необходимо добавить данные узлы в список no_access, выставив им флаг Sensor.

Например, для ограничения доступа к Telnet следует выполнить следующие действия:

Необходимо отредактировать файл /etc/xinetd.d/telnet и изменить строку flags следующим образом:

```
flags = SENSOR
deny_time = 30
disable = no
```

В результате доступ будет ограничен на 30 мин.

2.10 Определение открытых портов

Для определения открытых портов рекомендуется использовать сканер портов, например, nmap. Следующая команда, запущенная с консоли, определяет, какие порты ждут TCP-соединений из сети:

```
nmap -sT -O localhost
```

Чтобы проверить, не связан ли открытый порт с какой-либо известной службой, необходимо ввести:

```
cat /etc/services | grep <номер порта>
```

Чтобы проверить порт с помощью netstat, нужно выполнить следующую команду:

```
netstat -anp | grep <номер порта>
```

Далее необходимо убедится, что netstat указывает Pid использующего порт процесса, в этом случае угроза атаки через данный порт минимальна.

2.11 Использование брандмаэуров

Использование брадмауэров позволяет повысить защищенность ОС от угрозы сетевых вторжений.

2.11.1. Использование iptables

Запуск iptables осуществляется с помощью команды:

```
service iptables start
```

Чтобы служба iptables смогла работать, следует выключить IP6Tables, выполнив следующие команды:

```
service ip6tables stopchkconfig ip6tables off
```

Чтобы iptables по умолчанию запускалась при загрузке системы, необходимо изменить уровень выполнения этой службы с помощью chkconfig.

```
chkconfig --level 345 iptables on
```

2.11.2. Основные политики брандмауэра

Утверждение базовых политик брандмауэра создает основу для построения более подробных, определяемых пользователем правил. Для создания правил по умолчанию в iptables используются политики (–P). Обычно применяют политику «отбрасывания» всех пакетов и задают разрешающие правила только для каждого конкретного случая. Следующие правила блокируют все входящие и исходящие пакеты:

```
iptables P INPUT DROP
iptables P OUTPUT DROP
```

Кроме этого рекомендуется, чтобы все пересылаемые пакеты (пакеты, маршрутизируемые брандмауэром к точке назначения) также были запрещены — это защитит внутренних клиентов от нежелательного влияния Интернета. Для этого необходимо добавить следующее правило:

```
iptables P FORWARD DROP
```

Определив цепочки политики, возможно создавать новые правила для конкретной сети и требований безопасности. В следующих разделах рассматриваются некоторые правила, которые можно внедрить, настраивая брандмауэр iptables.

2.11.3. Сохранение и восстановление правил iptables

Правила брандмауэра существуют только пока компьютер включен, после перезагрузки правила автоматически сбрасываются и очищаются. Чтобы сохранить правила, чтобы они загрузились впоследствии, выполните команду:

```
/sbin/service iptables save
```

2.11.4. Использование механизмов фильтрации

Чтобы пользователи выполняли связанные с сетью функции и использовали сетевые приложения, необходимо открыть определенные порты. Например, чтобы разрешить доступ к 80-му порту средствами брандмауэра, необходимо добавить следующее правило:

```
iptables A INPUT p tcp m tcp sport 80 j ACCEPT
iptables A OUTPUT p tcp m tcp dport 80 j ACCEPT
```

Это позволит просматривать веб-содержимое сайтов, работающих на порту 80. Чтобы открыть доступ к защищенным веб-сайтам (https), необходимо также открыть порт 443.

```
iptables A INPUT p tcp m tcp sport 443 j ACCEPT
iptables A OUTPUT p tcp m tcp dport 443 j ACCEPT
```

Политика FORWARD позволяет администратору управлять тем, как пакеты маршрутизируются в локальной сети. Например, чтобы разрешить маршрутизацию для всей локальной сети (предположим, что брандмауэру/шлюзу назначен внутренний IP–адрес на интерфейсе eth1), можно задать такие правила:

```
iptables A FORWARD i eth1 j ACCEPT
iptables A FORWARD o eth1 j ACCEPT
```

Чтобы входящие HTTP-запросы маршрутизировались к выделенному HTTP-серверу с IP-адресом 10.0.4.2 (вне локальной сети 192.168.1.0/24), NAT обращается к таблице PREROUTING и передает пакеты по назначению:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT \  --to-destination
10.0.4.2:80
```

Для блокировки нестандартных портов, не используемых легитимными службами можно воспользоваться командами:

```
iptables -A OUTPUT -o eth0 -p tcp --dport 31337 --sport 31337 -j DROP
iptables -A FORWARD -o eth0 -p tcp --dport 31337 --sport 31337 -j DROP
```

Примечание: Брандмауэр IPv6 реализует подсистема Netfilter 6 и команда ip6tables. Для запуска службы используется команда:

```
service ip6tables start
```

3 Selinux

Пакет selinux при установке интегрируется в ядро ОС, по умолчанию в RELS SELinux не активирован.


Помните: SELinux может существенно повысить безопасность ОС, тем не менее мы не рекомендуем его использовать, если у администратора нет соответствующего опыта и перед ним не стоят задачи по обеспечению безопасности критически важной информации или сервисов. Однако в том случае, если сервера имеют доступ к Интернету, мы рекомендуем освоить принципы SELinux и обязательно его использовать.


Примечание: для корректной работы selinux в случае, если RELS был установлен в базовой конфигурации, может потребоваться установка дополнительных пакетов.

Настройка режимов работы selinux осуществляется в конфигурационном файле SELinux: /etc/selinux/config.

Режимы работы:

1) Включение SELinux:
   – в конфигурационном файле изменить строку:

`SELINUX=enforcing`

– перезагрузить компьютер.

2) Отключение SELinux:

– в конфигурационном файле изменить строку:

`SELINUX=disabled`

– перезагрузить компьютер

3) Режим предупреждений SELinux

– в конфигурационном файле изменить строку:

`SELINUX=permissive`

– перезагрузить компьютер

4) Включение SELinux в тестовом режиме:

– ввести в консоли `semanage`, Selinux будет запущен

– перезагрузить компьютер, настройки Selinux будут сброшены, Selinux отключен.

Примечание: для использования `semanage`, вероятно, необходимо будет установить дополнительные пакеты. Используйте команду:

```
yum -y install policycoreutils-python
```

Подробнее о `semanage` можно прочитать, если набрать `man semanage`.

3.1 Основные понятия Selinux

1) *Домен* — список того, что может делать процесс, или какие действия процесс может выполнять над различными типами. Домен однозначно определяет привилегии процесса.

2) *Тип* — список того, что можно сделать с объектом, т. е. список, определяющий привилегии доступа к объекту.

3) *Роль* — определяет список доменов, к которым имеет доступ пользовательская роль.

4) *Контекст безопасности* — это набор всех атрибутов, связанных с объектами типа файлов, каталогов, процессов, TCP сокетов и т.п.

5) *Политики* — это наборы правил, контролирующие такие списки ролей, к которым имеет доступ пользователь, какие роли имеют доступ к каким доменам и какие домены имеют доступ к каким типам.

3.2 Использование средств Selinux

3.2.1. Сопоставление пользователей Linux и Selinux

Для просмотра сопоставления пользователей введите команду:

```
semanage login -l
```

Вы увидите что-то вроде:

```
 Имя входа              Пользователь SELinux      Диапазон MLS/MCS
__default__             unconfined_u              s0-s0:c0.c1023
root                    unconfined_u              s0-s0:c0.c1023
system_u                system_u                  s0-s0:c0.c1023
test                    staff_u                   s0-s0:c0.c1023
useruser                user_u                    s0
```

Пользователи RELS по умолчанию будут сопоставлены с логином `__default__` SELinux (который в свою очередь сопоставлен пользователю `unconfined_u` SELinux). Если вы создадите нового пользователя командой `useradd`, не указав дополнительные опции, то он сопоставится пользователю SELinux `unconfined_u`).

3.2.2. Изменение сопоставлений по умолчанию

Как мы писали выше, пользователи RELS по умолчанию сопоставляются с логином `__default__` SELinux (который в свою очередь сопоставлен пользователю `unconfined_u` SELinux). Если необходимо создавать новых пользователей, не сопоставленных с ограниченным пользователем `unconfined_u`, можно изменить сопоставление по умолчанию командой:

```
semanage login -m -S targeted -s "user_ur s0 __default__
```

Новые пользователи RELS будут сопоставляться с `user_u`.

Проверить это можно командой:

```
semanage login -l
```

Вы увидите что-то вроде:

```
 Имя входа              Пользователь SELinux      Диапазон MLS/MCS


__default__             user_u                    s0
root                    unconfined_u              s0-s0:c0.c1023
system_u                system_u                  s0-s0:c0.c1023
test                    staff_u                   s0-s0:c0.c1023
useruser                user_u                    s0
```

3.2.3. Установка контекста безопасности по умолчанию для пользователей

Установка контекста безопасности производится в файле `/etc/selinux/targeted/contexts/default_contexts`, например,

`system_r:local_login_t user_r:user_t`.

3.2.4. Утилиты для выполнения действий с отличными от пользовательских правами

Утилита runcon

Утилита runcon позволяет выполнить произвольную команду с указанным контекстом безопасности. Синтаксис команды runcon:

```
runcon [-t <тип>] [-l <уровень>] [-u <пользователь>] [-r <роль>] <команда> [<аргументы>]
```

Утилита su

Утилита su используется, когда требуются привилегии суперпользователя. Она устраняет необходимость в прямой авторизации под учетной записью root.

Файл /etc/securetty содержит список терминалов, в которых может авторизоваться root. По умолчанию в этот список входят только локальные виртуальные консоли (tty).

Утилита sudo

Утилита sudo позволяет пользователю использовать его собственный пароль для доступа с привилегиями root к ограниченному набору команд. Это позволяет пользователю, например, монтировать съемные носители и открывать лоток устройства чтения компакт-дисков, не имея других привилегий root. Утилита sudo также ведет журнал всех успешных и неуспешных попыток выполнить команду sudo, что дает возможность отследить, кто и с какой целью выполнял эти команды.

4 Контексты Selinux

В ОС с запущенным SELinux, все процессы и файлы маркированы (помечены) так, чтобы представлять информацию в контексте безопасности. Эта информация называется контекстом SELinux, и просматривается с использованием команды ls -Z:

```
$ ls Z file1
-rw-rw-r-- user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

В этом примере, SELinux приводит пользователя (unconfined_u), роль (object_r), тип (user_home_t) и уровень (s0). Эта информация используется для принятия решений контроля доступа. В системах DAC (дискреционного контроля доступа), контроль доступа осуществляется с использованием ID пользователя и группы. Правила политики SELinux проверяются после правил DAC. Правила политики SELinux не используются, если DAC блокировал доступ первым.

Существует множество команд для управления контекстом SELinux для файлов, такие как chcon, semanage fcontext и restorecon.

4.1 Временные изменения: chcon

Команда chcon вносит изменения в контекст SELinux для файлов. Однако, изменения, вносимые с помощью команды chcon не сохраняются после перемаркирования ФС или выполнения команды /sbin/restorecon. Политика SELinux контролирует может ли пользователь изменять контекст для файлов. При использовании команды chcon, пользователи предоставляют всю информацию или часть об изменении контекста SELinux.

Некорректный тип файла обычно является причиной блокирования доступа SELinux.

В примерах, приведенных ниже показывается изменение атрибута тип контекста SELinux.

Пример: Выполнить команду cd без аргументов для перехода в домашний каталог.

Выполнить команду touch file1 для создания нового файла. Используя команду ls -Z file1 просмотрите контекст SELinux для file1:

```
$ ls -Z file1
-rw-rw-r- user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

В этом примере контекст SELinux для file1 включает пользователя SELinux unconfined_u, роль object_r, тип user_home_t и уровень s0.

Пример: Выполнить команду chcon -t samba_share_t file1 для изменения типа на samba_share_t. Опция -t отвечает за изменение типа. Просмотрите изменения с помощью ls -Z file1:

```
$ ls -Z file1
-rw-rw-r-- user1 group1 unconfined_u:object_r:samba_share_t:s0 file1
```

Пример: Использовать команду /sbin/restorecon -v file1 для восстановления контекста SELinux для файла file1. Использовать опцию -v для просмотра изменений, вносимых командой:

```
$ /sbin/restorecon -v file1
restorecon reset file1 context unconfined_u:object_r:samba_share_t:s0>system_u:
object_r:user_home_t:s0
```

В этом примере предыдущий тип samba_share_t восстанавливается на корректное значение, user_home_t. Когда используется целевая политика targeted (Политика SELinux по умолчанию), команда /sbin/restorecon читает файлы в каталоге /etc/selinux/targeted/contexts/files/, для того, чтобы узнать какие контексты SELinux должны присваиваться файлам.

4.2 Постоянные изменения: semanage fcontext

Команда `/usr/sbin/semanage fcontext` изменяет контекст SELinux для файлов. При использовании целевой политики `targeted`, изменения, вносимые этой командой, добавляются в файл `/etc/selinux/targeted/contexts/files/file_contexts`, если изменения вносятся для существующих файлов, то они добавляются в файл `file_contexts`, или добавляются в файл `file_contexts.local` для новых файлов и каталогов, например, при создании каталога `/web/`.

`Setfiles`, использующаяся при маркировании ФС, и `/sbin/restorecon`, использующаяся для восстановления контекста SELinux по умолчанию, читают эти файлы. Это значит, что изменения, вносимые командой `/usr/sbin/semanage fcontext` постоянны, даже если ФС будет перемаркирована. Политика SELinux контролирует возможность пользователей изменять контекст файлов.

В следующем примере демонстрируется как изменить тип файла в контексте SELinux.

Пример:

От имени пользователя `root` выполнить команду: `touch /etc/file1`

для создания нового файла. По умолчанию, вновь созданные файлы в каталоге `/etc/` помечаются типом `etc_t`:

```
# ls -Z /etc/file1
-rw-r--r-- root root unconfined_u:object_r:etc_t:s0 /etc/file1
```

От имени пользователя `root` выполнить команду:

`/usr/sbin/semanage fcontext -a -t samba_share_t /etc/file1`

для изменения типа файла `file1` на тип `samba_share_t`. Опция "`-a`" добавляет новую запись, а опция "`-t`" определяет тип (`samba_share_t`).

Примечание: Выполнение данной командой не изменяет тип напрямую, таким образом, `file1` остается помечен типом `etc_t`:

```
# /usr/sbin/semanage fcontext -a -t samba_share_t /etc/file1
# ls -Z /etc/file1
-rw-r--r-- root root unconfined_u:object_r:etc_t:s0 /etc/file1
```

Команда

`/usr/sbin/semanage fcontext -a -t samba_share_t /etc/file1`

добавляет следующую запись в файл:

```
/etc/selinux/targeted/contexts/files/file_contexts.local:
/etc/file1 unconfined_u:object_r:samba_share_t:s0
```

От имени пользователя `root` выполнить для изменения типа команду:

`/sbin/restorecon -v /etc/file1`

Так как команда semanage добавила запись в file.contexts.local для /etc/file1, команда /sbin/restorecon изменяет тип на samba_share_t:

```
# /sbin/restorecon -v /etc/file1
restorecon          reset          /etc/file1          context
unconfined_u:object_r:etc_t:s0>system_u:object_r:samba_share_t:s0
```

От имени пользователя root для удаления file1 выполнить команду:

```
rm -i /etc/file1
```

От имени пользователя root для удаления контекста, добавленного для /etc/file1, выполнить команду:

```
/usr/sbin/semanage fcontext -d /etc/file1
```

Когда контекст удален, выполнение команды restorecon изменяет тип на etc_t, вместо samba_share_t.

4.3 Монтирование ФС с изменением контекста

Для монтирования ФС с определенным контекстом, переопределяющим уже существующий контекст или указывающий другой контекст по умолчанию для ФС, не использующей расширенные атрибуты, от имени пользователя root для подключения нужной ФС используется команда:

```
mount -o context=SELinux_user:role:type:level
```

Изменения контекста не записываются на диск. По умолчанию, при монтировании NFS на стороне клиента, ФС присваивается контекст, назначенный политикой для ФС NFS. В большинстве политик этот контекст использует тип nfs_t. Без дополнительных опций монтирования, такой тип, может привести к отказам в доступе к ФС NFS для других служб, таких как Apache HTTP Server. Ниже показано как монтировать ФС NFS так, чтобы предоставлялся доступ для Apache HTTP Server:

```
# mount server:/export /local/mount/point o\
context="system_u:object_r:httpd_sys_content_t:s0"
```

Вновь созданные файлы и директории в этих ФС появляются с контекстом, указанным опцией -o context: однако, так как изменения контекста не записываются на диск, то для таких ситуаций, контекст, указанный опцией context сохраняется, только если опция context используется при следующем монтировании и с указанием того же самого контекста.

4.4 Создание постоянных контекстных монтирований

Для сохранения контекста постоянным при перемонтированиях и перезагрузках, необходимо добавить запись для ФС в /etc/fstab или в карту автомонтирования, и использовать нужный контекст как опцию монтирования. В следующем примере добавлена запись в /etc/fstab для монтирования NFS с нужным контекстом:

```
server:/export
/local/mount/ nfs context="system_u:object_r:httpd_sys_content_t:s0" 0 0
```

5 Сокращения и обозначения

DAC (Discretionary Access Control) — избирательное управление доступом

HTTP — HyperText Transfer Protocol

NFS (Network File System) — протокол доступа к ФС

RELS — ROSA Enterprise Linux Server

Selinux (Security-Enhanced Linux, Linux с улучшенной безопасностью) — реализация системы принудительного контроля доступа, реализован в ядре Linux

ОС — операционная система

ПО — программное обеспечение

ФС — файловая система