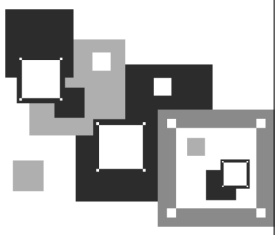
Глава 33



Система управления доступом SELinux

33.1. Что такое система управления доступом

В этой главе мы поговорим о SELinux — одной из самых популярных *систем управления доступом*. Кроме SELinux, существуют и другие системы управления доступом, например, GrSecurity и LIDS. Нужно отметить, что из этих систем самой строгой является SELinux — до сих пор она ни разу не была взломана (разумеется, при правильной ее настройке). Ясно, что все возможности SELinux мы не рассмотрим — они вполне заслуживают отдельной книги.

Для начала давайте разберемся, зачем нам нужна система управления доступом как таковая? В Linux есть две крайности, два типа пользователей: обычный пользователь и администратор (root). Права обычных пользователей можно ограничить и с помощью штатных средств Linux. Но если злоумышленник завладеет паролем администратора (как он это сделает — уже другой вопрос), то он получит полную власть над системой. Однако на компьютере с установленной системой управления доступом злоумышленник если и сможет выполнить какие-либо операции, то лишь те, которые не причинят системе ощутимого вреда.

Вернемся к обычным пользователям. Их права доступа ограничиваются, как правило, только доступом к файлам. Можно также задать ограничение на использование системных ресурсов: дискового пространства (квоты), процессорного времени, установить максимальное число процессов. И все. А система управления доступом может запретить пользователю выполнять те действия, которые он выполнять не должен. Например, зачем пользователю, который зарегистрировался в системе только для чтения почты, возможность компиляции исходного кода или запуска фоновых демонов? Теперь все становится на свои места — мы понимаем, что SELinux нужна.

Кроме всего прочего, SELinux контролирует и права доступа к файлам. Например, система проверила права доступа файла и разрешила доступ к какому-либо файлу. Но потом принимается за работу SELinux. Если в настройках SELinux указано, что данный пользователь (или процесс) не имеет доступа к этому файлу, то SELinux запрещает доступ к файлу, несмотря на имеющееся разрешение. И в самом деле, зачем Web-серверу доступ к каталогу /etc/selinux? Если же система запретила доступ к файлу (это первый этап — проверка прав доступа), тогда SELinux не задействуется.

33.2. Работаем с SELinux в Fedora и ASPLinux

Устанавливать SELinux мы будем не на "голый" компьютер, дабы не усложнять себе жизнь. Если вам нужна SELinux, установите дистрибутив Fedora или ASPLinux — в состав этих дистрибутивов SELinux входит по умолчанию.

Перейдите в каталог /etc/selinux. Там вы найдете файл config, управляющий настройками самой SELinux, а также каталог targeted, в котором будут находиться конфигурационные файлы политики targeted. В каталоге имеются три подкаталога: contexts, policy, users (контексты, политика, пользователи), а также файл booleans, в котором установлены некоторые булевы (логические) параметры. Пока эти файлы редактировать не нужно — оставьте все как есть.

Что же находится в каталогах contexts, policy и users? Чтобы получить ответ на этот вопрос, нужно обратиться к скучной теории.

- Начнем с базового понятия — понятия *сущности*. Сущность (identity) формирует часть контекста безопасности, задающего домены, в которые можно войти. То есть сущность определяет, что можно сделать. Не нужно путать сущность с идентификатором пользователя (UID). Они параллельно существуют в системе, но их смысл абсолютно разный. Обычно сущность представляется в системе так же, как и имя пользователя. Если в системе есть пользователь ppt и есть сущность ppt, выполнение команды `su` не изменяет сущности SELinux.

  Предположим, у нас есть пользователь ppt. Зарегистрируемся под его именем и выполним команду `id` (это команда SELinux), получим такой вывод:

  context=ppt:user_r:user_t

  Теперь введем команду `su`, наберем пароль root и снова введем команду `id`:

context=ppt:user_r:user_t

Мы получили тот же самый вывод. Контекст остался прежним и не изменился на контекст пользователя root. Правда, есть одно "но". Если сущности ppt разрешен доступ к роли `sysadm_r` (до сих пор роль `user_r`), и пользователь выполнит команду `newrole -r sysadm_r` (изменит свою роль), а потом снова выполнит команду `id`, то получит вывод:

context=ppt:sysadm_r:sysadm_t

Сущность осталась такой же, но роль и домен (второе и третье поле) изменились. Таким образом, сущность определяет, какие роли и домены могут быть использованы.

- *Домен* (domain) однозначно определяет привилегии процесса. Другими словами, домен представляет собой список того, что может сделать процесс, или, точнее, какие операции может выполнить процесс над разными типами.

  Примеры доменов: `sysadm_t` — домен администратора системы, `user_t` — домен для непривилегированных пользователей. Процесс init выполняется в домене `init_t`, а процесс named — в `named_t`.

- *Тип* (type) задается для объекта и определяет доступ к этому объекту. Практически, тип — это то же самое, что и домен, но если домен относится к процессам, то тип — к файлам, каталогам, сокетам и т. п.

- *Роль* (role) определяет список доменов, которые могут быть использованы. Домены, разрешенные для пользовательской роли, определяются в файлах политики. Если роль не имеет доступа к домену, то при попытке выполнения действия с доменом доступ будет запрещен.

  Лучше всего это продемонстрировать на примере: если вам нужно разрешить непривилегированным пользователям (домен `user_t`) выполнять команду `passwd`, в конфигурационном файле следует прописать:

  ```
  role user_r types user_passwd_t
  ```

  Из этой команды видно, что пользователь с ролью `user_r` может входить в домен `user_passwd_t`, то есть ему разрешено выполнять команду `passwd`.

- *Контекст безопасности* (security context) — это набор всех атрибутов, которые связаны с файлами, каталогами, процессами, TCP-сокетами. Контекст безопасности состоит из сущности, роли, домена (или типа вместо домена). Команда `id` выводит текущий контекст безопасности.

- *Решение о переходе* (transition) определяется контекстом безопасности, который будет назначен выполняемой операцией. Существуют два вида переходов:

- переход домена процесса — используется при выполнении процесса определенного типа;
- переход типа файла — используется при создании файла в определенных каталогах.

❒ Наконец, рассмотрим последнее понятие — понятие *политики*. Политика — это набор правил, контролирующих списки ролей, к которым у пользователя есть доступ, доступ ролей к доменам, доступ доменов к типам.

Зарегистрируйтесь в системе как пользователь root и введите команду:

```
# system-config-securitylevel
```

В окне **Настройка уровня безопасности** перейдите на вкладку **Настройка SELinux** (рис. 33.1). По умолчанию SELinux обычно выключена. Для ее включения установите флажок **Включено**.

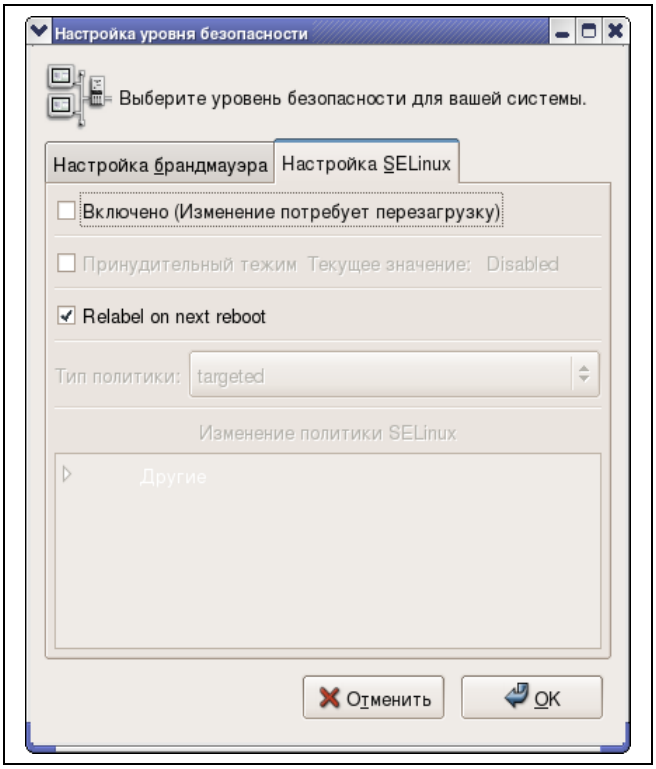


Рис. 33.1. Окно **Настройка уровня безопасности**

Сразу после установки этого флажка вы увидите предупреждение о необходимости перемаркировать файловую систему правильными контекстами безопасности (рис. 33.2). Перемаркировка будет выполнена после перезагрузки системы. В этом же окне сообщается, что перемаркировка (аналог команды `make relabel`) может занять довольно много времени (зависит от размера файловой системы).

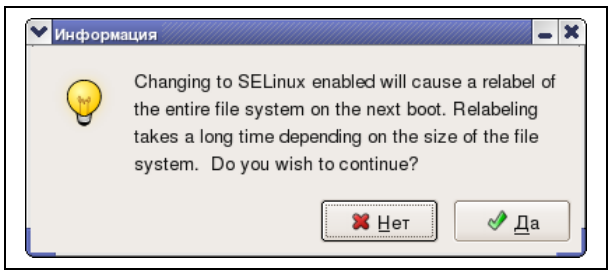


Рис. 33.2. Предупреждение о необходимости перемаркировки

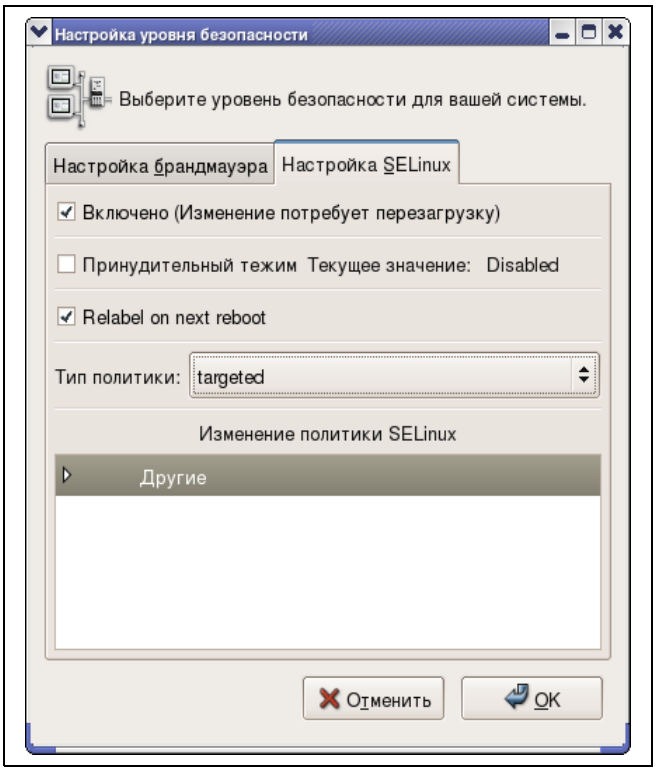


Рис. 33.3. SELinux включена

Теперь обратите внимание на изменившееся окно **Настройка SELinux** (рис. 33.3). В нем можно выбрать тип политики SELinux. Пока доступна только одна политика — **targeted** (целевая).

Нажмите на кнопку **OK** и перезагрузите компьютер командой `reboot`. В процессе старта системы появится сообщение:

Warning -- SELinux relabel is required *

Это сообщение свидетельствует о том, что SELinux будет перемаркировывать файловую систему. После этого пойдет загрузка — все как обычно. При входе в X Window на первую консоль будет выведено несколько не совсем обычных сообщений. Их формат мы разберем чуть позже. Первое, что хочется сделать — это ввести команду `id`, чтобы просмотреть свой контекст безопасности (рис. 33.4):

```
context=root:system_r:hotplug_t
```

Роль `system_r` — это роль системы, которая выше роли `sysadm_r`.

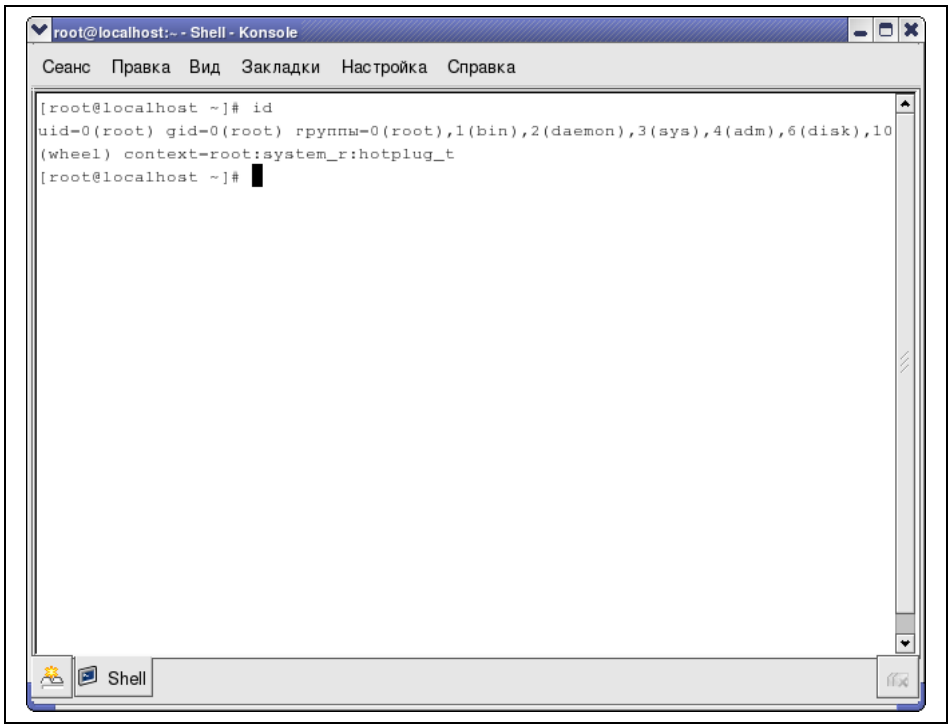


Рис. 33.4. Контекст безопасности пользователя root

Теперь самое время обратиться к конфигурационным файлам SELinux. Откройте файл /etc/selinux/config. В этом файле будут всего две директивы: `SELINUX` и `SELINUXTYPE`. Первая может принимать следующие значения:

- `enforcing` — применить политику безопасности SELinux;
- `permissive` — режим отладки, вместо запрета тех или иных операций SELinux будет просто выводить предупреждения;
- `disabled` — SELinux отключена.

Для второй директивы возможны два значения:

- `targeted` — защищаются только целевые сетевые демоны (которые будут явно указаны);
- `strict` — полная защита.

> ***Совет***
>
> Если вам нужна полная защита, установите пакет selinux-policy-strict, находящийся на первом диске ASPLinux 11. Со второго компакт-диска я бы посоветовал установить пакет selinux-doc — дополнительная документация никогда не помешает.

33.3. Аудит политик

Для аудита политик SELinux используется программа seaudit, но при запуске мы получаем сообщение, что не установлена политика по умолчанию. Самое интересное, что на дистрибутивных дисках я так и не нашел пакет policy, содержащий политику по умолчанию. Пришлось его взять из Интернета: **ftp://rpmfind.net/linux/ASPLinux/i386/RPMS.10/policy-1.11.3-3.noarch.rpm** (рис. 33.5).

33.4. Создание роли

Роль имеет большое значение — у каждой роли свои полномочия, например, у роли `sysadm_r` полномочий намного больше, чем у `user_r`. Поэтому нужно знать, как можно изменить роль. Вообще-то в цели **targeted**, в которой мы сейчас работаем, роли пользователей особого интереса не представляют, по-

скольку осуществляется защита только выбранных сетевых демонов, но о команде `newrole` сказать все-таки нужно.

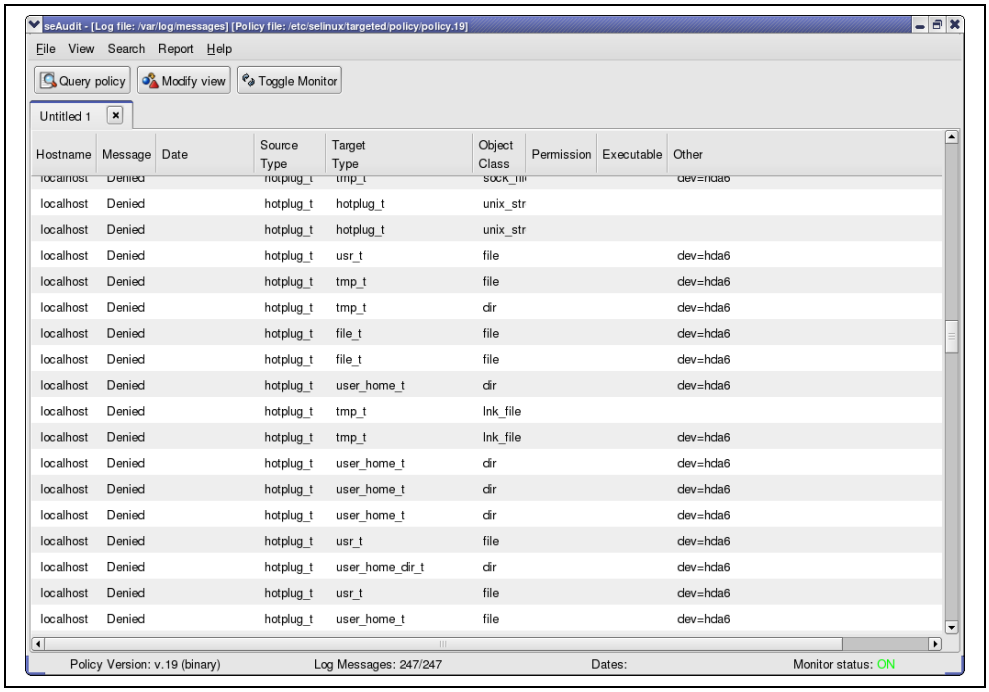


Рис. 33.5. Аудит политик

Ее синтаксис следующий:

```
newrole -r роль
```

Например:

```
newrole -r sysadm_r
```

После выполнения этой команды нужно будет ввести пароль для сущности (пароль пользователя). Если нет прав доступа к указанной роли, вы увидите сообщение:

ppt:sysadm_r:sysadm_t is not a valid context

Здесь указывается, что сущность `ppt` не имеет права доступа к роли `sysadm_r`.

33.5. Псевдофайловая система /selinux

При запуске системы с поддержкой SELinux в корне появится каталог /selinux — это псевдофайловая система SELinux (наподобие псевдофайловой системы /proc). С ее помощью можно изменять некоторые параметры — например, режим работы SELinux. Как уже было отмечено, есть два режима работы — разрешающий (permissive) и принудительный (enforcing). В первом режиме SELinux только "ругается", а ОС работает так же, как и обычная Linux-система без SELinux, а во втором — применяются все настроенные политики. Отладочные сообщения в разрешающем режиме протоколируются в файле /var/log/messages. Для переключения в принудительный режим используется команда:

```
echo "1" > /etc/selinux/enforce
```

Для перехода в разрешающий режим служит команда:

```
echo "0" > /etc/selinux/enforce
```

33.6. Пользователи и SELinux

Лучше добавить всех необходимых пользователей в систему до включения SELinux, но бывают случаи, когда сделать это просто невозможно (система работает продолжительное время, и все пользователи заведены). Если SELinux уже активна, то для добавления нового пользователя нужно выполнить следующие действия:

1. Становимся администратором: `$ su`.
2. Входим в роль `sysadm_r`:

   ```
   # newrole -r sysadm_r
   ```

3. Добавляем нового пользователя:

   ```
   # useradd -c "New user" -m -d /home/newuser -g users -s /bin/bash -u
   1005 newuser
   # passwd newuser
   ```

4. Но этого мало. Нужно еще настроить роли пользователя. Для этого в файл /etc/selinux/users добавляем строку:

   ```
   user newuser roles { user_r };
   ```

   Этим мы назначаем пользователю `newuser` роль `user_r`.

5. Если нужно, чтобы пользователь имел доступ к нескольким ролям, тогда укажите несколько ролей через пробел, например:

```
user setest roles { user_r sysadm_r };
```

6. Для активации изменений введите команду:

```
# make -C /etc/selinux load
```

Активация изменений займет некоторое время, после чего вы увидите такие сообщения:

Success
touch tmp/load
make: Leaving directory `/usr/share/selinux/policy/current'

Примечание

Следует отметить, что если пользователю нужен доступ только к роли `user_r`, это можно явно не указывать. Явно указывать роль надо лишь в том случае, когда пользователю требуется изменить свой пароль самостоятельно.

33.7. Конфигуратор system-config-securitylevel (system-config-selinux)

Теперь начинается самое интересное — мы будем редактировать нашу политику.

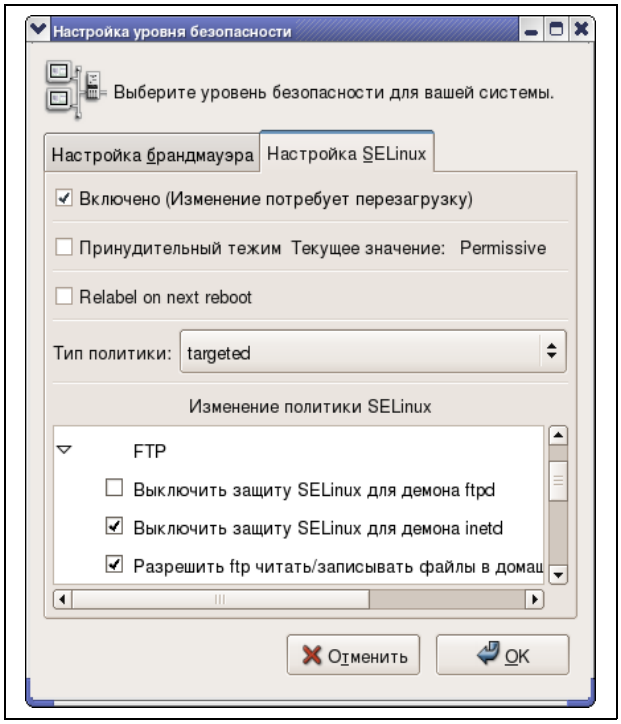


Рис. 33.6. Редактирование политики FTP

Сейчас у нас используется политика **targeted**, подразумевающая защиту только указанных нами сетевых демонов. Запустите конфигуратор system-config-securitylevel (или system-config-selinux). На вкладке **Настройка SELinux** (после активации SELinux) появится возможность редактирования политики. Там все просто: приводится список служб и для каждой службы — набор опций SELinux. Например, вот список опций для FTP (рис. 33.6):

- **Выключить защиту SELinux для демона ftpd**;
- **Выключить защиту SELinux для демона initd**;
- **Разрешить ftp читать/записывать файлы в домашних каталогах**.

А вот список привилегий пользователя (рис. 33.7):

- **Позволить пользователям читать любые файлы по умолчанию**;
- **Разрешить пользователям запускать pppd соединения**.

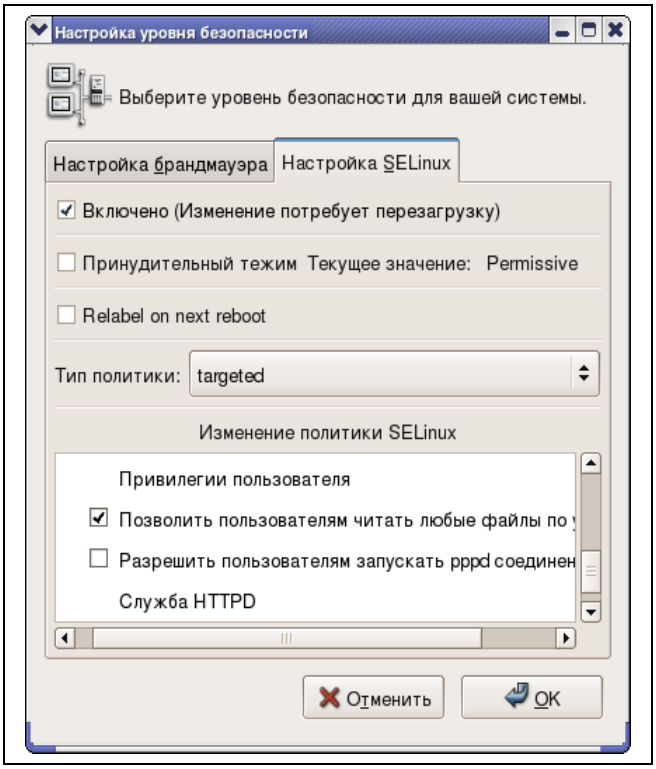


Рис. 33.7. Редактирование привилегий пользователей

Примечание

В новых версиях дистрибутива Fedora (7/8) для редактирования политик SELinux используется конфигуратор system-config-selinux.

33.8. Журналы SELinux

Рассмотрим пример типичного сообщения о нарушении доступа, которое можно обнаружить в файле /var/log/messages:

```
May 21 14:44:12 localhost kernel: audit(1148208252.610:29): avc: denied {
read } for pid=2054 comm="bash" name=".bash_profile" dev=hda6 ino=23695
scontext=root:system_r:hotplug_t tcontext=root:object_r:user_home_t
tclass=file
```

Фрагмент `avc: denied` означает, что операция была запрещена. Далее следует идентификатор процесса, пытающегося выполнить операцию (`for pid`), имя процесса (`comm`), имя объекта (`name`), имя устройства (`dev`), номер инода объекта (`ino`), контекст безопасности процесса (`scontext`), контекст безопасности объекта (`tcontext`, в данном случае это файл `.bash_profile`) и тип целевого объекта (`tclass=file`, тип объекта — файл).