

Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)

---

Факультет «Информатика и системы управления»  
Кафедра «Компьютерные системы и сети»

**В.Ю. Мельников**

# **Исследование методов защиты операционных систем и данных**

Электронное учебное издание

Методические указания по выполнению лабораторных работ  
по дисциплине "Операционные системы"

2019

# Введение

Цель работы - исследование методов защиты информации в Linux.

Продолжительность работы - 4 часа.

Linux изначально был разработан многопользовательским. В него были заложены средства защиты от случайного или преднамеренного повреждения данных, а так же защиты конфиденциальной информации. Со временем, появлялись новые средства защиты, но новые методы защиты действуют совместно с традиционной системой защиты, которая остаётся актуальной.

## Модели управления доступом

Напомню (обзорно) модели управления доступом, реализованные в Linux:

**Избирательное (дискреционное) управление доступом** ([discretionary access control, DAC](#)) заключается в том, что каждому объекту системы назначается список пользователей, причём для каждого из них задаётся список допустимых операций (читать, писать и т. д.). Каждый объект системы имеет привязанного к нему пользователя, называемого владельцем. Именно владелец устанавливает права доступа к объекту.

**Управление доступом на основе ролей** ([Role Based Access Control, RBAC](#)) - развитие политики избирательного управления доступом. Разрешения назначаются не отдельным пользователям, а группам пользователей со сходными полномочиями.

В традиционной системе, для каждого файла и каталога назначаются права на чтение, запись и исполнение: 1 пользователю-владельцу, 1 группе-владельцу и группе прочих пользователей. Несмотря на простоту, возможностей этой системы достаточно в большинстве случаев. Информацию именно этой системы отображают все файловые менеджеры. Кроме того, права пользователей настраиваются в конфигурационных файлах многих программ согласно документации на эти программы.

**Улучшенная система прав доступа** [ACL](#) позволяет назначить права произвольному количеству пользователей и групп.

**Мандатное управление доступом** ([Mandatory access control, MAC](#)) - разграничение доступа субъектов к объектам, основанное на назначении метки конфиденциальности для информации, содержащейся в объектах, и выдаче официальных разрешений (допуска) субъектам на обращение к информации такого уровня конфиденциальности. В отличие от дискреционной модели, владелец файла не имеет полной свободы назначения прав доступа к своему файлу.

# Управление пользователями и группами

## Субъекты прав доступа

Субъектами прав доступа в Linux являются пользователи и их группы.

При установке Linux мы вводили пароль суперпользователя. Суперпользователь всегда имеет UID = 0 и имя «root». Он имеет доступ ко всем ресурсам независимо от настроек доступа к ним. Именно поэтому не рекомендуется постоянно работать с учётной записи суперпользователя – в случае взлома сессии злоумышленник получит доступ ко всем ресурсам.

Программы и демоны часто запускаются от имени специальных пользователей. Обычно, эти пользователи имеют то же имя, что и программа и создаются автоматически при установке программы. Под именем этого пользователя нельзя войти, он не имеет домашнего каталога. В случае взлома программы, злоумышленник получит доступ только к файлам этой программы и общедоступным ресурсам.

Каждый пользователь может принадлежать к нескольким группам. Создание групп и внесение пользователя в группы выполняется суперпользователем (обычно при создании нового пользователя).

Далее, каждому файлу и каталогу назначаются различные права доступа для различных пользователей и групп.

## Создание пользователей

Добавим пользователя «user1» и зададим ему пароль:

```
useradd -m -s /bin/bash user1
passwd user1
<пароль>
<повтор пароля>
```

Опция «-s /bin/bash» задаёт интерпретатор команд (shell). В debian, по умолчанию используется sh, а он гораздо слабее чем bash.

Не забудьте опцию -m. Она создаёт домашний каталог, записывает в него шаблоны сценариев инициализации. По умолчанию, создаётся домашний каталог /home/ИМЯ\_ПОЛЬЗОВАТЕЛЯ. Посмотрим, его содержимое:

```
ls -la /home/user1
```

```
root@debian:~# ls -la /home/user1
итого 20
drwxr-xr-x 2 user1 user1 4096 окт 29 22:48 .
drwxr-xr-x 4 root  root  4096 окт 29 22:48 ..
-rw-r--r-- 1 user1 user1  220 окт 29 22:48 .bash_logout
-rw-r--r-- 1 user1 user1 3515 окт 29 22:48 .bashrc
-rw-r--r-- 1 user1 user1  675 окт 29 22:48 .profile
```

Команда автоматически создала следующие сценарии:

~/.bash\_profile — выполняется при входе пользователя в систему;

~/.bashrc — выполняется при каждом запуске дочернего интерпретатора команд;

~/.bash\_logout — выполняется при выходе из системы.

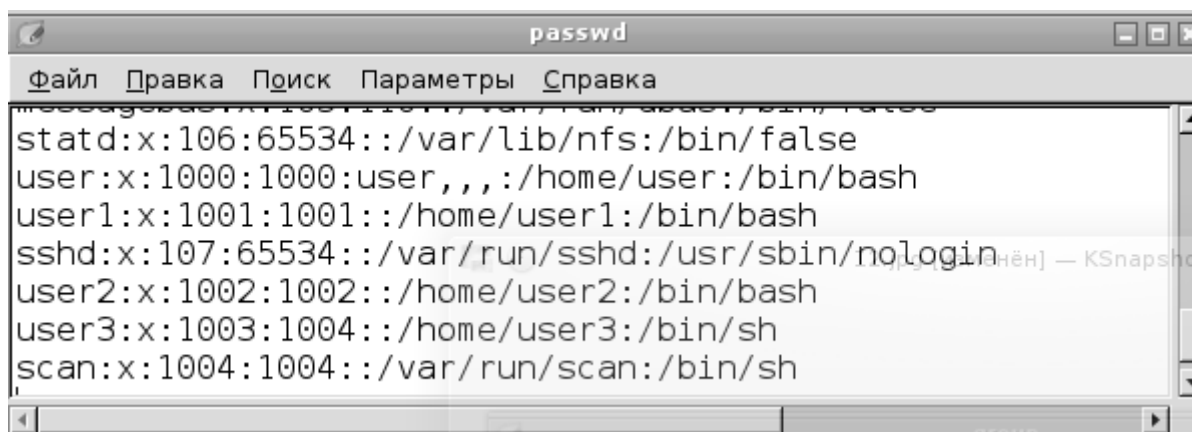
В файле «.bashrc» пользователь может добавить:

- Алиасы команд с часто употребляемыми опциями. Например, «alias ll='ls -l'». Этой командой мы сегодня часто будем пользоваться. Раскомментируйте соответствующую строку.
- Зададим вид приглашения командной строки. «PS1="\[\033[31m\]\u@\h\[\033[m\]:\w\[\033[33;1m\]\\$ \[\033[m\] » теперь имя пользователя (\u) имя компьютера (\h) будут выводиться красными символами на чёрном фоне (\033[31m), текущий каталог (\w) обычным шрифтом (\033[m) и символ \$ желтым цветом (\033[33;1m). Смотрится очень красиво. Такое приглашение полезно задать в файле /root/.bashrc, чтобы сразу было видно, что мы подключились как суперпользователь и одна неверная команда может привести к фатальным последствиям. Если часто приходится подключаться удалённо к нескольким серверам, полезно раскрасить имена этих компьютеров в разные цвета, чтобы не перепутать тестовый сервер с «боевым». (таблицу цветов и подробности можно посмотреть на странице <http://rus-linux.net/nlib.php?name=/MyLDP/consol/color-ru.html>)
- команда umask определяет права на вновь создаваемые файлы. Мы рассмотрим её позже

Домашние каталоги пользователей создаются в каталоге «/home». Отдельно лежит домашний каталог пользователя «root» он находится непосредственно в корневом каталоге. Если в «/root» нет файла «.bashrc» - скопируйте из домашнего каталога любого пользователя.

Чтобы изменения вступили в силу запустите новый экземпляр интерпретатора команд командой «bash»

Учётные всех пользователей данные хранятся в файле /etc/passwd:



Разберите по документации (например, «man 5 passwd») и перечислите в отчёта поля

этого файла. Обратите внимание, что несмотря на «говорящее» название, пароли в этом файле не хранятся.

Изменить информацию о пользователе можно прямо в этом файле (командой `vi pw`), но слишком легко зацепить соседних пользователей, поэтому лучше использовать команду `usermod`

Для удаления пользователей используется команда:

`userdel ПОЛЬЗОВАТЕЛЬ`

Для смены пароля используется команда:

`passwd ПОЛЬЗОВАТЕЛЬ`

В отличие от прочих команд этой темы эта команда доступна всем пользователям. Так что пароль пользователя может сменить как `root`, так и сам пользователь.

## **Группы пользователей**

Если несколько пользователей должны иметь доступ к файлу или каталогу следует создать группу и предоставить права этой группе.

При создании пользователя командой `useradd -m` автоматически создаётся так же группа с именем пользователя. Этой группой можно воспользоваться, чтобы дать права на свой файл только одному пользователю.

Команда `groupadd ГРУППА` создаёт новую группу

Команда `groupdel ГРУППА` удаляет группу.

Команда `usermod -g ГРУППА ПОЛЬЗОВАТЕЛЬ` - изменяет первичную группу пользователя. Файлы и каталоги, создаваемые пользователем будут принадлежать этой первичной группе. При создании пользователя в `debian`, пользователю автоматически назначается группа с тем же именем, что и имя пользователя.

Команда `usermod -G ГРУППА1,ГРУППА2,... ПОЛЬЗОВАТЕЛЬ` - задаёт список дополнительных групп пользователя. **ВНИМАНИЕ!** При этом старый список дополнительных групп теряется, поэтому обычно используют другую форму:

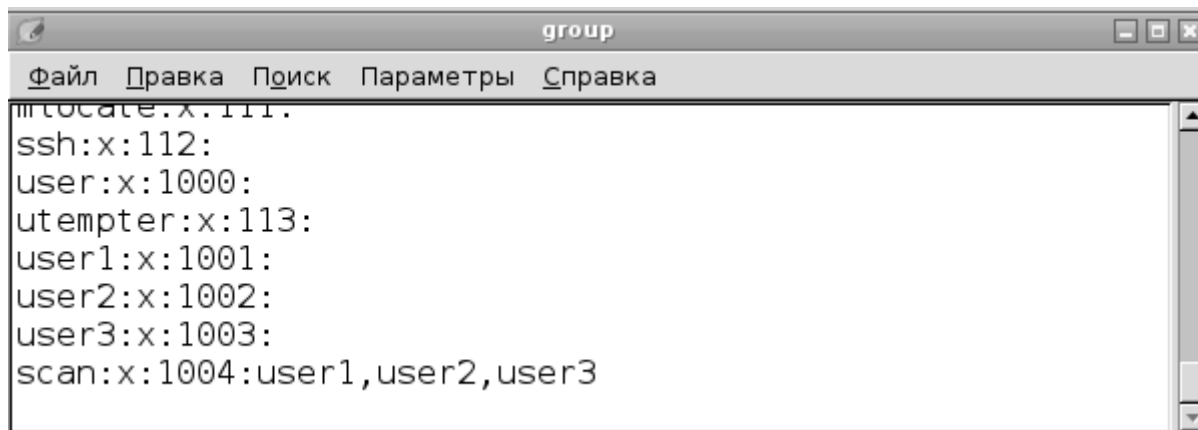
Команда `usermod -a -G ГРУППА ПОЛЬЗОВАТЕЛЬ` - добавляет к списку групп заданного пользователя заданную дополнительную группу.

**ВНИМАНИЕ!** На уже запущенные программы (в том числе интерпретатор команд) изменения не действуют. Можно открыть новое окно консоли, но достаточно дать команду `bash` чтобы запустить новый экземпляр интерпретатора команд.

Для просмотра, в какие группы входит пользователь, используется команда `groups ПОЛЬЗОВАТЕЛЬ`.

```
root@debian:/home# groupadd grp1
root@debian:/home# usermod -a -G grp1 user1
root@debian:/home# groups user1
user1 : user1 grp1
root@debian:/home# usermod -a -G grp1 user
root@debian:/home# groups user
user : user cdrom floppy audio dip video plugdev netdev grp1
root@debian:/home# _
```

Полный список групп с атрибутами содержится в файле «/etc/group».



Этот файл можно редактировать (командой «vigr») но слишком легко зацепить соседних пользователей, поэтому лучше использовать команду «usermod».

Чтобы просмотреть полный список групп используйте команду

```
cat /etc/group
```

Список пользователей, входящих в группу можно отфильтровать командой:

```
grep ГРУППА: /etc/group
```

## Традиционная система прав доступа к файлам и каталогам

### Назначение прав на файл

Пользователь, который создаёт файл или каталог становится его владельцем. И имеет на него все права.

Чтобы дать доступ к этому файлу другим пользователям следует:

- Создать группу (groupadd)
- Включить в неё этих пользователей. (usermod)
- Сменить у файла группу — владельца (chgrp)
- При необходимости, дать права на запись (chmod)
- Пользователь «root» может ещё сменить владельца (chown), но прочим пользователям эта операция запрещена.

Если надо дать права только одному пользователю, можно воспользоваться группой, которая автоматически создаётся вместе с пользователем и имеет имя пользователя.

## Определение существующих прав

Вернёмся к выводу команды

```
ls -la /home/user1
```

```
root@debian:~# ls -la /home/user1
итого 20 1 3 4
drwxr-xr-x 2 user1 user1 4096 окт 29 22:48 .
drwxr-xr-x 4 root root 4096 окт 29 22:48 ..
-rw-r--r-- 1 user1 user1 220 окт 29 22:48 .bash_logout
-rw-r--r-- 1 user1 user1 3515 окт 29 22:48 .bashrc
-rw-r--r-- 1 user1 user1 675 окт 29 22:48 .profile
```

В 3 колонке отображается пользователь-владелец (user1)

В 2 колонке отображается группа-владелец (user1). В нашем случае это первичная группа пользователя. Её имя совпадает с именем пользователя.

В 1 символе 1 колонки отображается признак каталога (d)

Далее в первой колонке следуют три тройки символов (rwx), отражающие права соответственно: пользователя, группы и прочих пользователей. Если прав на одно из трёх действий нет, в соответствующей позиции стоит «-»

В нашем примере, для файлов отображается «-rw-r--r--». Это расшифровывается так:

- «-» - это не каталог
- «rw-» - Пользователь владелец имеет права на чтение(r) и запись(w). Прав на исполнение (x) нет — третьим символом стоит «-»
- «r--» - Группа владелец имеет права только на чтение(r)
- «r--» - Прочие пользователи имеют права только на чтение(r)

А для каталогов в первой колонке отображается «drwxr-xr-x». Это расшифровывается так:

- «d» - признак каталога
- «rwx» - Пользователь владелец имеет права на чтение(r) и запись(w) файлов в каталог. Бит «x» для каталогов означает права на доступ в каталог. Можно читать и выполнять файлы из каталога, даже если нет прав на чтение самого каталога. И наоборот, если нет прав на доступ в каталог, даже если у каталога стоит бит «r», команда «ls» покажет пустой каталог и доступа к файлам каталога и его подкаталогам не будет.
- «r-x» - Группа владелец имеет права на чтение(r) и доступ в каталог
- «r-x» - Прочие пользователи имеют права на чтение(r) и доступ в каталог (x)

В приведённом примере нет файлов с битом «x», но для файлов это признак прав на выполнение. Большинство файлов в каталоге «/usr/bin» имеют права «rwxr-xr-x».

Кроме признаков «r», «w», «x» имеются ещё несколько полезных признаков прав. Эти признаки выводятся в соответствующей тройке вместо признака «x».

| Категории пользователей      | Символ | Для файла                                                                                        | Для каталога                                                                                                                                                                          |
|------------------------------|--------|--------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Права пользователя владельца | s      | Любой пользователь может запустить файл с правами владельца<br>Применяется для системных утилит. | Не применяется                                                                                                                                                                        |
| Права группы                 | s      | Любой пользователь может запустить файл с правами группы<br>Применяется для системных утилит.    | Все файлы, создаваемые в каталоге принадлежат группе, владеющей каталогом<br>Применяется для общих каталогов группы                                                                   |
| Права прочих                 | t      | Не применяется                                                                                   | Удалять и переименовывать файлы в этом каталоге может только владелец файла или каталога, даже если есть права на запись в этот каталог<br>Применяется для каталогов временных файлов |

## Изменение прав на файлы и каталоги

Разберём права на домашние каталоги пользователей полученные при создании:

```
root@debian:/home# ls -l /home/
итого 8
drwxr-xr-x 2 user user 4096 ноя 18 02:15 user
drwxr-xr-x 7 user1 user1 4096 ноя 25 01:51 user1
```

У каталога «/home/user1»

Пользователь-владелец – «user1», группа-владелец – его собственная группа «user1». «d» признак каталога. Права пользователя-владельца (первая тройка) - «rwx» (естественно все права).

А вот права группы-владельца и прочих пользователей, пожалуй, избыточны: (вторая и третья тройки) «r-x» – права на чтение каталога (r) и возможность обращаться к подкаталогам и файлам этого каталога (x). С этими же правами пользователь будет создавать все новые каталоги и файлы. Значит, если ничего не предпринять, любой пользователь сможет просмотреть любой каталог и файл пользователя.

Безопаснее было бы установить права «d rwx --- ---»

С другой стороны, если пользователь хочет создать в домашней папке каталог и открыть к нему доступ для некоторой группы, придётся дать хотя бы права «d rwx --x ---», иначе пользователи этой группы не дойдут до этого каталога. Но лучше создавать общедоступные каталоги в каталоге «/usr/share»



Чтобы дать права на удаление из каталога надо установить бит «w». Прав на сам файл не требуется.

Права на файл и каталог можно изменить командой:

**chmod [-R] РЕЖИМ ФАЙЛ**

Если задана опция -R, команда меняет права не только заданного каталога, но и входящих в него файлов и каталогов (рекурсивно). Обратите внимание, «R» надо задавать именно в верхнем регистре.

Режим задаётся в форме «[ugoa][+ -=][rwxst]»

Первая группа символов определяет категорию пользователей: владелец (u), группа (g), прочие пользователи (o), все категории (a)— определяет

Вторая группа определяет воздействие: разрешить (+), запретить (-) установить (=)

Третья группа символов определяет изменяемые права (смотри приведённую выше таблицу)

Рассмотрим некоторые примеры:

`chmod o-rw /home/user1` — отнимает у «прочих» пользователей права просматривать и изменять домашний каталог пользователя user1

```
root@debian:/home/user1# ls -l /home/
итого 8
drwxr-xr-x 2 user  user  4096 ноя 18 02:15 user
drwxr-xr-x 7 user1 user1 4096 ноя 25 01:51 user1
root@debian:/home/user1# chmod o-rw /home/user1
root@debian:/home/user1# ls -l /home/
итого 8
drwxr-xr-x 2 user  user  4096 ноя 18 02:15 user
drwxr-x--x 7 user1 user1 4096 ноя 25 01:51 user1
```

`chmod o+x /home/user1` — даёт «прочим» пользователям возможность добраться до некоторых файлов и подкаталогов домашнего каталога пользователя user1, даже когда у них нет прав на чтение самого каталога.

`chmod g=r /home/user1` — даёт права только на чтение пользователям группы, которой принадлежит файл.

`chmod u=rwx,go-rwx /home/user1/testdir` даёт все права владельцу файла и отнимает их у группы и прочих

`chmod a-rwx /home/user1/testfile` отнимает все права у всех, включая владельца. Теперь доступ к testfile имеет только пользователь root. Пользователь root вообще имеет права на всё что угодно. Постарайтесь не дать команду «rm -rf /» от имени rot. Она удаляет все файлы и каталоги в файловой системе. Поэтому, подключайтесь пользователем root только когда вы собираетесь администрировать систему

Опытные пользователи иногда задают в команде «chmod» права в виде восьмеричного числа.

Число составляется из следующих битов:

| Владелец             | Группа               | Прочие               |
|----------------------|----------------------|----------------------|
| 400(r) 200(w) 100(x) | 040(r) 020(w) 010(x) | 004(r) 002(w) 001(x) |

Например:

`chmod 700 /home/user1/testdir` — даёт все права (7=4+2+1) пользователю владельцу и отнимает все права у группы и прочих пользователей.

`chmod 777 /home/user1/testdir` — даёт все права для всех.

`chmod 644 file` — даёт права на чтение и запись (6=4+2) пользователю владельцу и права на чтение (4) у группы и прочих.

Рука тянется дать «`chmod -R 644 КАТАЛОГ`», чтобы дать права только на чтение для всех файлов каталога с подкаталогами. Вот только после этой команды даже владелец потеряет доступ к файлам каталога. Дело в том, что этой командой мы сбросим бит «x» у заданного каталога и его под каталогов. В результате, никто не может зайти в каталог чтобы прочесть файл на который права то, есть. Команда «`chmod -R 755 КАТАЛОГ`» тоже не годится. Так мы дадим права на выполнение всем подряд файлам.

Правильной будет команда «`chmod -R go-w КАТАЛОГ`» - отнять права на запись у группы и прочих пользователей.

Другой способ — выполнить 2 команды:

```
find КАТАЛОГ -type f -exec chmod 644 {} \;  
find КАТАЛОГ -type d -exec chmod 755 {} \;
```

Первая команда ищет, начиная с заданного каталога все файлы (-type f) и выполняет для каждого из них команду «`chmod 644`»

Вторая команда ищет все каталоги (включая заданный) и назначает права на них.

## Смена владельца

Для того, чтобы изменить группу, которой принадлежит файл или каталог следует воспользоваться командой:

`chgrp [-R] ГРУППА ФАЙЛ`

При необходимости, пользователя - владельца файла можно сменить командой:

`chown [-R] ВЛАДЕЛЕЦ ФАЙЛ`

Если задана опция -R, команда меняет права не только заданного каталога, но и входящих в него файлов и каталогов (рекурсивно).

Эту команду часто приходится давать, если файл или каталог был создан пользователем root (или, программой, запущенной от имени root). Но операция смены владельца считается опасной, поэтому позволена только пользователю «root».

Можно одной командой сменить и пользователя — владельца и группу:

**chown [-R] ВЛАДЕЛЕЦ:ГРУППА ФАЙЛ**

```
root@debian:/home# mkdir /home/user1/work
root@debian:/home# ls -l /home/user1/
итого 4
drwxr-xr-x 2 root root 4096 фев 11 00:30 work
root@debian:/home# chown user1:user1 /home/user1/work/
root@debian:/home# ls -l /home/user1/
итого 4
drwxr-xr-x 2 user1 user1 4096 фев 11 00:30 work
```

Иногда, этой же командой изменяют только группу

**chown [-R] :ГРУППА ФАЙЛ**

## ***Права доступа на новые файлы и каталоги***

При создании новых файлов и каталогов, в том числе путём копирования, новый файл получает права, заданные командой

**umask РЕЖИМ**

Чтобы узнать текущий режим дайте эту команду без параметров. Выводится восьмеричное число, которое указывает какие права доступа **запрещены**:

Число составляется из тех же битов, что в команде «chmod» (но смысл обратный):

| Владелец             | Группа               | Прочие               |
|----------------------|----------------------|----------------------|
| 400(r) 200(w) 100(x) | 040(r) 020(w) 010(x) | 004(r) 002(w) 001(x) |

Данная команда не задаёт права на выполнение файлов.

По умолчанию установлено «umask=022» — запрещено изменение файлов и каталогов группе и прочим пользователям. Остаются права на чтение файлов и каталогов.

Режим «umask=027» запрещает изменение файлов и каталогов группе и запрещает все операции прочим пользователям.

## ***Смена пользователя (switch user)***

Когда требуется сменить пользователя используйте команду

**su ПОЛЬЗОВАТЕЛЬ**

Но помните, что при такой смене пользователя не выполняются сценарии инициализации, и вы остаётесь в том же каталоге.

Для того, чтобы получить полное окружение заданного пользователя используйте команду

**su -l ПОЛЬЗОВАТЕЛЬ**

Для смены пользователя на root удобнее дать команду «su» без параметров (в некоторых версиях linux «su -»)

Чтобы вернуться к работе с прежним пользователем, выполните команду «exit»

Воспользуйтесь командой «pstree» и определите, как работает команда «su» и что делает «exit».

Имя текущего пользователя можно посмотреть командой «whoami»

Если забыли эту команду, можно воспользоваться командой «id».

```
user1@debian95:~$ id
uid=1001(user1) gid=1001(user1) группы=1001(user1),27(sudo)
```

Она выдаёт имя не только имя пользователя, но и группы, в которые он входит.

## **Выполнение команд от имени другого пользователя**

Помните, что суперпользователь root имеет права, на абсолютно все операции и работать под этим именем надо как можно меньше.

Для того, чтобы выполнить одну команду с правами другого пользователя безопаснее использовать одну из следующих команд:

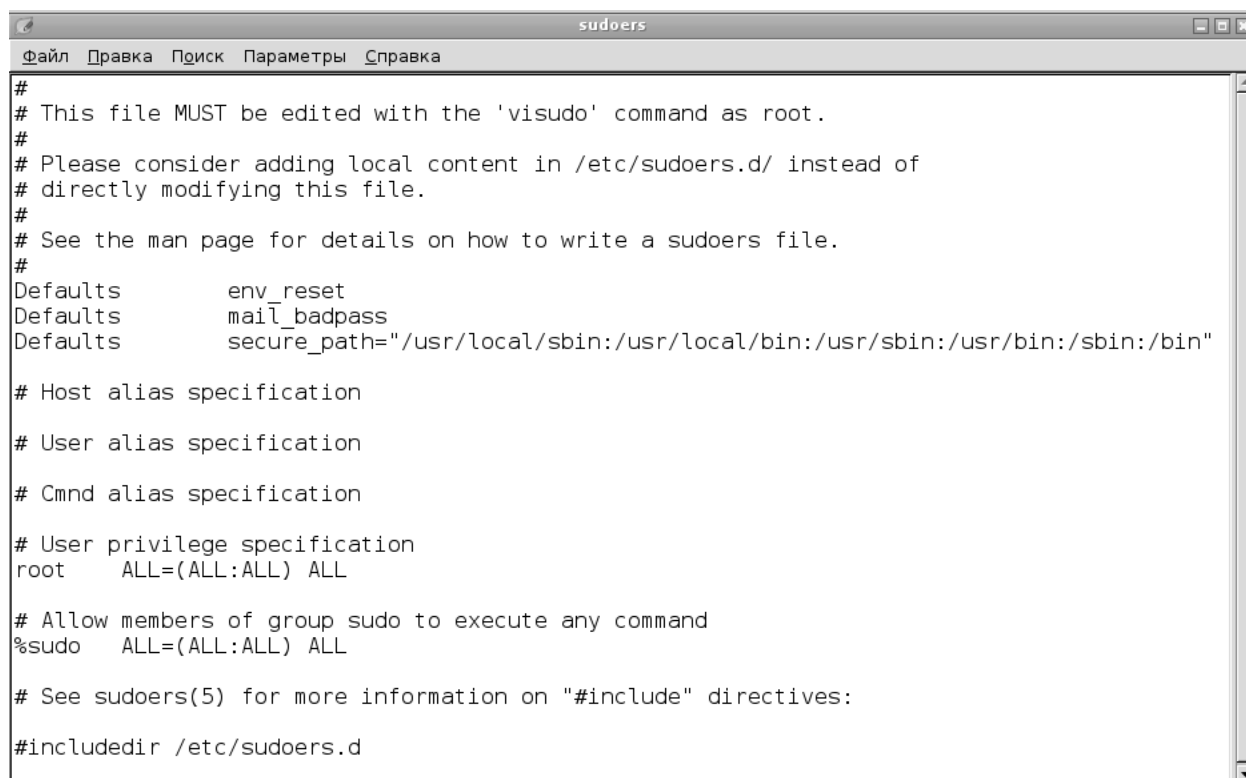
**sudo -u ПОЛЬЗОВАТЕЛЬ КОМАНДА**

**sudo КОМАНДА** (для выполнения команды от имени root)

sudo (обычно расшифровывается как «superuser do») – запуск команд от имени суперпользователя. Во многих дистрибутивах эта утилита является предустановленной, но в используемых в ЛР облегчённых дистрибутивах её нет. Установить её можно следующей командой:

```
apt-get install sudo
```

Настройки полномочий пользователей заданы в файле «/etc/sudoers».

A screenshot of a text editor window titled 'sudoers'. The window shows the contents of the /etc/sudoers file. The text includes comments about editing the file with 'visudo', default settings for env\_reset, mail\_badpass, and secure\_path, host and user alias specifications, user privilege specifications for root and the sudo group, and a directive to include /etc/sudoers.d.

```
sudoers
Файл  Правка  Поиск  Параметры  Справка
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "#include" directives:
#include /etc/sudoers.d
```

Можно добавить строку:

```
user1 ALL=(ALL) ALL
```

Но правильнее добавить пользователя в группу «sudo» командой:

```
usermod -a -G sudo user1
```

На уже запущенные программы (в том числе интерпретатор команд) изменения не

действуют. Можно открыть новое окно консоли, но достаточно дать команду «bash» чтобы запустить новый экземпляр интерпретатора команд.

Убедитесь, что теперь можно устанавливать недостающие пакеты без смены пользователя:

```
sudo apt-get update
```

Добавьте в «/etc/sudoers» строку:

```
Defaults timestamp_timeout=0,rootpw
```

У sudo есть таймаут работы (в минутах), в течение которых sudo не будет просить повторно ввести пароль. «**timestamp\_timeout=0**» делает таймаут нулевым. Теперь sudo будет спрашивать пароль при каждом запуске.

По умолчанию, sudo запрашивает пароль текущего пользователя. Указание «rootpw» - заставит утилиту запрашивать пароль пользователя «root».

## **Улучшенная система прав доступа (ACL)**

В большинстве случаев, достаточно традиционной системы прав доступа, в сложных случаях в Linux имеется улучшенная система [ACL](#) (Access Control List — список контроля доступа). В частности, с помощью неё можно задать на заданный файл, или каталог разные права для каждого из нескольких пользователей и/или групп.

Начиная с Debian 9 для использования ACL сначала надо установить пакет «acl» командой «apt-get install acl»

Создадим файл с именем «file» командой «touch file»

Команда «getfacl file» выводит полную информацию о правах доступа к файлу или каталогу.

```
root@debian95:~$ touch file
root@debian95:~$ getfacl file
# file: file
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

Для установки прав доступа используется утилита «setfacl».

Чтобы дать пользователю «user1» все права на файл «file» надо дать команду

```
setfacl -m "u:user1:rwx" file
```

Чтобы дать группе «grp1» все права на файл «file» надо дать команду

```
setfacl -m "g:grp1:r-x" file
```

```
root@debian95:~$ getfacl file
# file: file
# owner: root
# group: root
user::rw-
user:user1:rwx
group::r--
group:grp1:r-x
mask::rwx
other::r--
```

## Системы мандатного контроля доступа

Традиционная, дискреционная модель безопасности (DAC) неплохо справляется с обеспечением безопасности только при условии ответственного отношения пользователя. Пользователь имеет полную свободу действий над своими файлами и может намеренно или случайно открыть доступ к секретным файлам.

В модели DAC процессы выполняются от имени пользователей, которые их запустили и при этом получают все права этого пользователя. Если процесс запускает root, процесс получает права абсолютно на любую операцию.

Мандатные системы доступа (MAC) предназначены для того, чтобы сократить последствия ошибок в коде и настройках сервисов, а так же ошибок пользователей. В MAC пользователь, обладающий мандатом некоторого уровня, имеет доступ к ресурсам более низкого уровня доступа (т.е. менее защищённым), но не имеет доступа к более защищённым. Пользователь не может полностью управлять правами доступа к создаваемым им ресурсам во избежание проникновения злоумышленников.

А ещё MAC дают возможность настройки прав процессов без создания специальных пользователей для каждого процесса. И запротоколировать попытки несанкционированного доступа.

В Linux Мандатная система доступа реализуется средствами SELinux, AppArmor.

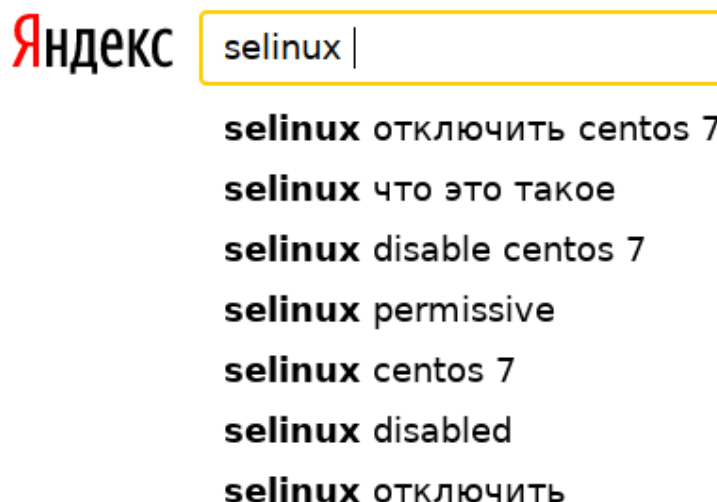
В [Astra Linux](#), разработанной для нужд Минобороны РФ используется собственная разработка. В ней даже буфер обмена не передаётся на более высокий уровень, и на снимке экрана, окна приложений, работающих на более высоком уровне доступа закрашены.

На данный момент, наиболее широко распространена SELinux. Рассмотрим её.

# SELinux

## Зачем нужна SELinux

Если набрать в поисковике «SELinux», выпадает:



Первый совет «гуру» при любой проблеме «попробуй отключить SELinux»

### ОДНАКО!

Согласно требованиям ФСТЭК мандатное управление доступом или «метки доступа» являются ключевым отличием систем защиты Государственной Тайны РФ старших классов 1В и 1Б от младших классов защитных систем [2]. Поддержка систем мандатного доступа давно встроена в ядро Linux. В многих релизах Linux, SELinux устанавливается и активируется при инсталляции.

В CentOS и многих других релизах, SELinux устанавливается по умолчанию.

Так что пользоваться придётся.

## **Установка, включение и отключение SELinux**

ВНИМАНИЕ! в Debian 10 перед установкой SELinux надо удалить apparmor:

```
apt-get remove apparmor  
reboot
```

Для установки SELinux надо дать команду:

```
apt-get install selinux-basics selinux-policy-default auditd
```

Добавляем запуск SELinux в конфигурационный файл GRUB командой

```
selinux-activate
```

Далее придётся перезагрузиться

```
reboot
```

Даём команду «check-selinux-installation» для проверки правильности

установки.

```
root@debian:~# check-selinux-installation
Traceback (most recent call last):
  File "/usr/sbin/check-selinux-installation", line 33, in <module>
    results += test.test()
  File "/usr/share/selinux-basics/tests/24_fsckfix.py", line 24, in test
    raise IOError("/etc/default/rcS not found, is this Debian?")
OSError: /etc/default/rcS not found, is this Debian?
root@debian:~# _
```

Обнаружена одна ошибка. Каталог «rcS» в очередной версии куда то переместили.

Остальные тесты прошли.

Основные настройки Selinux хранятся в файле «/etc/selinux/config»

```
root@debian:~# cat /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
# default - equivalent to the old strict and targeted policies
# mls      - Multi-Level Security (for military and educational use)
# src      - Custom policy built from source
SELINUXTYPE=default

# SETLOCALDEFS= Check local definition changes
SETLOCALDEFS=0
root@debian:~# _
```

Режим работы задаётся настройкой «SELINUX=РЕЖИМ

| Режим      | Описание                                                        |
|------------|-----------------------------------------------------------------|
| disabled   | Полностью отключён                                              |
| permissive | Только фиксировать нарушения                                    |
| enforcing  | Блокируются действия, не соответствующие политикам безопасности |

Посмотреть текущий режим можно командой «getenforce» или «sestatus»

```
root@debian95:~$ sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:      /etc/selinux
Loaded policy name:           default
Current mode:                 permissive
Mode from config file:       permissive
Policy MLS status:           enabled
Policy deny_unknown status:   allowed
Max kernel policy version:    30
```

Сейчас SELinux находится в режиме «permissive». В этом режиме он только записывает подконтрольные операции протокол в файл «/var/log/audit/audit.log».

В режиме «enforcing» SELinux тоже ведёт протокол, но не только фиксирует, но и блокирует недопустимые операции. Если какой-то сервис не работает, загляните в файл протокола и поищите слово «denied»



Можно временно переключить SELinux в режим «enforcing» командой «setenforce 1», и обратно в режим «permissive» командой «setenforce 0». После перезагрузки снова будет действовать режим, заданный в файле «/etc/selinux/config». Чтобы защита автоматически активировалась при перезагрузке задав в файле «/etc/selinux/config» «SELINUX=enforcing». Но не делайте этого, пока в «/var/log/audit/audit.log» появляются записи о блокировании операций. Есть шанс, что debian вообще не загрузится.

Совсем отключить SELinux можно только задав в файле «/etc/selinux/config» «SELINUX=disabled» и перезагрузив компьютер.

Ну вот. Отключать SELinux вы уже умеете. Однако, лучше научиться разрешать только то, что вам нужно.

## **Политики безопасности**

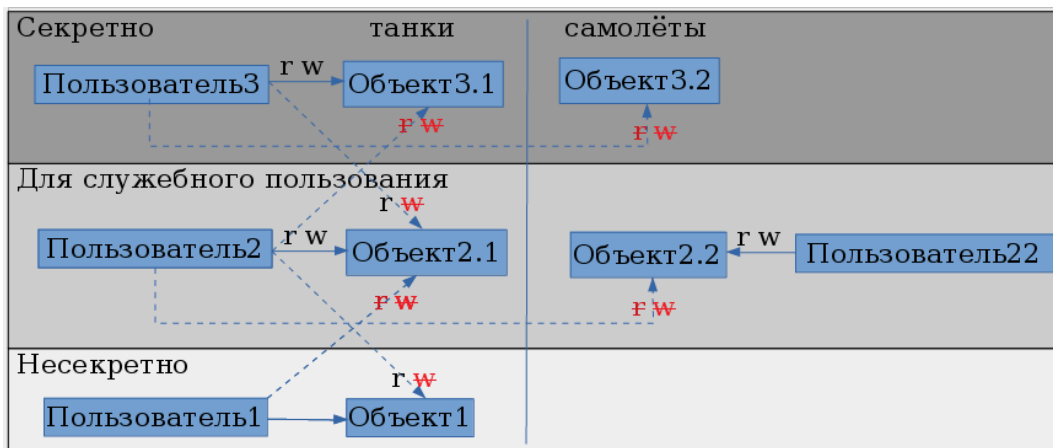
В системе SELinux по умолчанию все операции запрещены. Правила политики безопасности разрешают часть операций. Чтобы облегчить системным администраторам работу, были разработаны стандартные политики безопасности. Они содержатся в пакете «selinux-policy-default»

Политика безопасности задаётся в файле «/etc/selinux/config» настройкой «SELINUXTYPE=ПОЛИТИКА»

В Debian 9 можно выбрать следующие политики:

| Политика | Описание                                                                                                                                                                                                                                                                                                                       |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| default  | Политика типа TE ( <a href="#">Type Enforcement</a> ) / RBAC (Roles Based Access Control) — доступ согласно типам / ролям.<br>Ограничивает доступ только для нескольких широко распространенных служб. Сеансы пользователей не ограничены, и поэтому маловероятно, что SELinux заблокирует законные пользовательские операции. |
| mls      | Политика типа MLS ( <a href="#">Multi-Level Security</a> ) / MCS (Multi Categories Security)<br>Содержит не только правила, защиты основных служб, но и определяет уровни безопасности.                                                                                                                                        |
| src      | Политики безопасности пользователя, сформированные из исходников.                                                                                                                                                                                                                                                              |

Рассмотрим сначала **MLS/MS**:



Пользователи, могут читать только документы своего уровня доступа и ниже. Кроме того, пользователи могут записывать информацию только в объекты своего уровня. Так они не смогут передать секретные сведения пользователям нижележащего уровня. Причём, метки доступа включаются в сетевые пакеты, так, что и копирование на другой компьютер не поможет.

Кроме уровней доступа метки безопасности содержит категории, которые определяют, предметную область (например, виды вооружений). Даже на своём уровне безопасности пользователь может не иметь доступа к объектам другой категории.

Но **MLS/MS** касается в основном военных, а правила политик безопасности вида **Type Enforcement (TE)** широко применяются. Во многих дистрибутивах Linux SELinux устанавливается при установке. При работающем SELinux не обязательно создавать нового пользователя для каждого сервиса, достаточно определить правила в политике безопасности. Тогда сервис или приложение смогут получить доступ только к необходимым им файлам и каталогам. Например Skype читает файл `/etc/passwd` и каталог `~/.mozilla`. Конечно паролей в файле «passwd» нет, а из `~/.mozilla` Skype читает только настройки проксирования, но кто его знает, куда он ещё полезет. Исходных кодов разработчик не публикует.

Важно что можно определить не только права на чтение и запись файлов, но и на выполнение различных операций. Для одной и той же команды можно разрешить только часть операций.

### Как проверяются права доступа?

- Сначала проверяются традиционные права доступа (DAC). SELinux и другие системы безопасности дополняют DAC
- Затем начинает работать LSM (Linux Security Modules). Это универсальный интерфейс, благодаря которому возможно использование различных систем

безопасности (SELinux, AppArmor, Tomsy и Smack.)

- Все обращения к объектам системы передаются в SELinux или другую систему безопасности. В запросе передаются объект, субъект и контекст безопасности. Контекст безопасности (или метка безопасности) это строка, которая записывается в атрибуты файла. В этой строке перечислены: пользователь, роль, тип данных, уровень и категория безопасности.
- SELinux ищет в политиках безопасности соответствующее правило. Если правило найдено, доступ разрешается, иначе запрещается.
- Все действия протоколируются в файлах каталога «/var/log/audit».

## Контекст безопасности SELinux

Дайте команду «id»

```
root@debian95:~$ id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

После установки SELinux появилась информация о контексте безопасности:

«контекст=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023»

Контекст безопасности содержит информацию:

- «unconfined\_u» (неограниченный) - пользователь SELinux. В SELinux свои пользователи;
- «unconfined\_r» (неограниченный)- роль;
- «unconfined\_t» (неограниченный) - домен — именно к домену привязываются правила политик безопасности;
- «s0-s0» - доступные уровни безопасности;
- «c1023» - категория безопасности.

По умолчанию, процессы, запущенные пользователем наследуют его контекст безопасности. Дайте команду:

ps -Z

| LABEL                                                 | PID | TTY   | TIME     | CMD  |
|-------------------------------------------------------|-----|-------|----------|------|
| unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 | 624 | pts/0 | 00:00:00 | bash |
| unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 | 630 | pts/0 | 00:00:00 | ps   |

Мы видим, что процессы, bash и ps унаследовали метку безопасности от пользователя.

Но многие программы помечаются специальной меткой и получают собственный домен. Дайте команду:

ls -Z /usr/sbin/ntpd\*

```
system_u:object_r:ntpd_exec_t:s0 /usr/sbin/ntpd
```

Мы видим, что программа ntpd имеет тип «ntpd\_exec\_t».

Дайте команду «ps axu -Z | grep ntpd» и укажите в отчёте, от имени какого пользователя запущен процесс и какую метку безопасности получил.

Этот механизм назначения метки программе позволяет предоставить программе только те права, которые ей требуются. В традиционной схеме для этого приходилось создавать специального пользователя.

Обратите внимание, что, с опцией «-Z» традиционные команды linux выводят метку безопасности SELinux.

## Пользователи SELinux

В SELinux собственные пользователи. Чтобы получить таблицу соответствия традиционных пользователей linux пользователям SELinux дайте команду:

[semanage login](#) -l

| Имя входа   | Пользователь SELinux | Диапазон MLS/MCS | Служба |
|-------------|----------------------|------------------|--------|
| __default__ | unconfined_u         | s0-s0:c0.c1023   | *      |
| root        | unconfined_u         | s0-s0:c0.c1023   | *      |
| system_u    | system_u             | s0-s0:c0.c1023   | *      |

Пользователю root соответствует пользователь «unconfined\_u»;

Пользователь «system\_u», от имени которого работают сервисы;

Прочие пользователи (\_\_default\_\_) работают от имени «unconfined\_u».

В традиционном linux, пользователь включается в группы и так приобретает права. В SELinux пользователю назначаются одна или несколько ролей. Дайте команду:

[semanage user](#) -l

| Пользователь SELinux | Разметка Префикс | MLS/MLS/ Уровень | MCS MCS Диапазон | Роли SELinux              |
|----------------------|------------------|------------------|------------------|---------------------------|
| root                 | sysadm           | s0               | s0-s0:c0.c1023   | staff_r sysadm_r system_r |
| staff_u              | staff            | s0               | s0-s0:c0.c1023   | staff_r sysadm_r          |
| sysadm_u             | sysadm           | s0               | s0-s0:c0.c1023   | sysadm_r                  |
| system_u             | user             | s0               | s0-s0:c0.c1023   | system_r                  |
| unconfined_u         | unconfined       | s0               | s0-s0:c0.c1023   | system_r unconfined_r     |

Укажите в отчёте роли традиционного пользователя «root»

В большинстве случаев администратор должен использовать «semanage login». Пользователи SELinux в основном определяются базовой политикой и обычно не требует модификации. Перевод справки по команде «[semanage](#)» приведён в приложении.

## Настройки безопасности сервисов

Если разработчики SELinux правильно настроили политики безопасности, SELinux не мешает работе и вспоминаешь о нём, только когда требуется сделать нестандартные настройки сервисов. Установим WEB сервер «apache» и рассмотрим на его примере как разрешить ему использовать нестандартные параметры.

Установим WEB сервер «[apache HTTP server](#)»

apt-get install apache2

В настройках задан стандартный каталог для страниц сайта «/var/www/html/».

Запишем в него файл простенькой страницы

```
echo '<!DOCTYPE HTML><html><body><h1>Hello World</h1>
</body></html>' >/var/www/html/index.html
```

Перейдите теперь в браузере перейти по адресу «http://АДРЕС» (подставьте адрес вашей виртуальной машины) вы увидите «Hello World» крупным жирным шрифтом. Пока всё работает.

Пусть нам требуется на одном физическом сервере разместить два сайта сторонних разработчиков. Если на страницах сайта заданы относительные ссылки, то такие сайты можно разместить в двух подкаталогах каталога «/var/www/html/» - «/var/www/html/site1» и «/var/www/html/site2». Тогда сайты окажутся расположены по адресам «http://АДРЕС/site1» и «http://АДРЕС/site2». Но достаточно часто, в коде на javascript или в коде PHP страниц заданы ссылки относительно корня сайта. Одним из путей решения этой проблемы является использование разных http портов. Например, 80 (стандартный) и 81. Соответственно адреса сайтов будут: «http://АДРЕС» и «http://АДРЕС:81».

Часто для формирования динамических страниц нужны файлы данных, которые должны храниться в определённых каталогах. Надо научиться настраивать к ним доступ поэтому, для тренировки, пусть страницы второго сайта будут лежать у нас в каталоге «/srv/www/site2». Запишем страницу второго сайта в каталог «/srv/www/site2»

```
mkdir -p /srv/www/site2
echo '<!DOCTYPE HTML><html><body><h1>This is a site2</h1>
</body></html>' >/srv/www/site2/index.html
```

Теперь добавим конфигурационный файл второго сайта

```
nano /etc/apache2/sites-enabled/site2.conf
```

Вставляем в него текст:

```
Listen 81
<VirtualHost *:81>
    DocumentRoot /srv/www/site2
    ErrorLog ${APACHE_LOG_DIR}/error_site2.log
    CustomLog ${APACHE_LOG_DIR}/access_site2.log combined
    <Directory /srv/www/site2>
        Require all granted
    </Directory>
</VirtualHost>
```

Сохраняем (Ctrl+o). Выходим из редактора (Ctrl+x).

Рестартуем WEB-сервис:

```
systemctl restart apache2
```

И пытаемся перейти в браузере на адрес «http://АДРЕС:81»

Вы увидите «This is a site2» крупным жирным шрифтом. Пока всё работает. Если не работает — загляните в файл «/var/log/error\_site2.log»

Включаем режим защиты

```
setenforce 1
```

Рестартуем WEB-сервис:

```
systemctl restart apache2
```

Выдаётся ошибка.

Смотрим ошибки доступа в файле «/var/log/audit/audit.log»:

```
grep -E 'denied.*permissive=0' /var/log/audit/audit.log
```

```
type=AVC msg=audit(1561230340.020:219): avc: denied { sys_resource } for pid=2549
comm="apachectl" capability=24 scontext=system_u:system_r:httpd_t:s0 tcontext=system
_u:system_r:httpd_t:s0 tclass=capability permissive=0
type=AVC msg=audit(1561230340.420:221): avc: denied { sys_resource } for pid=2556
comm="apachectl" capability=24 scontext=system_u:system_r:httpd_t:s0 tcontext=system
_u:system_r:httpd_t:s0 tclass=capability permissive=0
type=AVC msg=audit(1561230340.624:222): avc: denied { name_bind } for pid=2559 com
m="apache2" src=81 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r:r
eserved_port_t:s0 tclass=tcp_socket permissive=0
type=AVC msg=audit(1561230340.632:223): avc: denied { name_bind } for pid=2559 com
m="apache2" src=81 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r:r
eserved_port_t:s0 tclass=tcp_socket permissive=0
```

Начинаем с последней записи

**avc: denied** - операция запрещена

**{ name\_bind }** - Наименование операции, в данном случае –  
связывание сетевого сокета с локальным адресом

**for pid=2559 comm="apache2"** - программа

**src=81** – в данном случае TCP-порт=81

**scontext=system\_u:system\_r:httpd\_t:s0** – контекст безопасности  
процесса

**tcontext=system\_u:object\_r:reserved\_port\_t:s0** – контекст  
безопасности объекта

**tclass=tcp\_socket** – класс объекта – в данном случае TCP сокет.

**Permissive=0** – режим работы – в данном случае не permissive –  
значит enforcing

Вывод: Надо разрешить WEB-сервисам доступ к TCP-порту 81 командами:

```
setenforce 0
```

```
semanage port -a -t http_port_t -p tcp 81
```

```
setenforce 1
```

Тип объекта «http\_port\_t» указан в [документации](#) на Debian

Снова рестартуем WEB-сервис:

```
systemctl restart apache2
```

Не этот раз ошибок нет.

Пытаемся перейти в браузере на страницу «http://АДРЕС:81». На этот раз выводится  
сообщение «Forbidden».

Опять SELinux что то блокирует

Смотрим ошибки доступа в файле «/var/log/audit/audit.log»:

```
grep -E 'denied.*permissive=0' /var/log/audit/audit.log
```

```
type=AVC msg=audit(1561233168.523:379): avc: denied { getattr } for pid=2615 comm="apache2" path="/srv/www/site2/index.html" dev="sda1" ino=159907 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:var_t:s0 tclass=file permissive=0
```

Снова смотрим последнюю запись:

```
type=AVC msg=audit(1561233168.523:379):
avc: denied - операция запрещена
{ getattr }
for pid=2615 comm="apache2"
path="/srv/www/site2/index.html" - путь к файлу
dev="sda1" ino=159907
scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:var_t:s0
tclass=file - класс объекта – в данном случае файл
permissive=0 – режим работы – в данном случае не permissive –
значит enforcing
```

Вывод: Надо разрешить доступ WEB-сервисам к каталогу «/srv/www/» командами:

```
setenforce 0
semanage fcontext -a -t httpd_sys_content_t "/srv/www(/.*)"?"
restorecon -R /srv/www/
setenforce 1
```

Команда `semanage fcontext` — добавляет новое правило безопасности

Команда `restorecon` — сбрасывает кэш.

Команды приведены в [документации](#) на Debian.

## Заключение

Дайте команду:

```
grep -E 'denied.*permissive=0' /var/log/audit/audit.log
```

И подключитесь к вашей виртуальной машине по протоколу ssh

Посмотрите сообщения в первом окне:

```
type=AVC msg=audit(1561235461.254:515): avc: denied { rename } for pid=2792 comm="sshd" name="motd.dynamic.new" dev="tmpfs" ino=24451 scontext=system_u:system_r:sshd_t:s0-s0:c0.c1023 tcontext=system_u:object_r:var_run_t:s0 tclass=file permissive=0
type=AVC msg=audit(1561235461.262:516): avc: denied { read } for pid=2792 comm="sshd" name="motd.dynamic" dev="tmpfs" ino=15818 scontext=system_u:system_r:sshd_t:s0-s0:c0.c1023 tcontext=unconfined_u:object_r:var_run_t:s0 tclass=file permissive=0
```

Несмотря на то, что подключение прошло успешно, часть операций блокируется. Что то связанное с временными файлами. Где то это может аукнуться.

Рекомендую, в Debian устанавливать и использовать SELinux с осторожностью. Сначала тестировать всё устанавливаемое программное на «стенде» - тестовом сервере, максимально приближенном к основному серверу.

И вообще, если вам действительно нужен SELinux, лучше воспользоваться релизами, в которых SELinux по умолчанию устанавливается при инсталляции. В этих релизах все программы тестируются в условиях активизированного SELinux и проблем обычно не возникает.

## Задание

Предположим, в вашей организации в нескольких подразделениях стоят сканеры, пользователи сканируют на них документы и записывают на сервер. На сервере работает программа, которая распознаёт файлы из одного каталога и записывает файлы с распознанным текстом в другой каталог.

Пользователи загружают распознанные тексты документов в систему документооборота, кроме документов «для служебного пользования», которые записывают в специальный каталог. К этому каталогу имеют доступ только определённая группа пользователей.

**Задача: С помощью команд, не пользуясь ACL и SELinux!**

- Создать несколько пользователей, включая пользователя от имени которого работает сервис распознавания.
- Для каждого пользователя создать каталоги:
  - in — для файлов, предназначенных для распознавания
  - out — для распознанных файлов
- Настроить доступ к файлам и каталогам. Пользователи не должны иметь доступ к файлам других пользователей. Не забудьте дать права сервису распознавания.
- Создать каталог, в который выкладывают файлы пользователи группы «DSP». Только пользователи этой группы должны иметь к нему доступ.
- Создать файл протокола, в который записывает сообщения сервис распознавания. Все пользователи должны иметь права на чтение этого файла.

## Приложения

Текст этого раздела представляет собой перевод фрагментов справки из команды «man», относящихся к материалу данного учебного пособия. Данный материал даёт расширенную информацию по рассмотренным командам, а так же может использоваться для выполнения заданий, приведённых в тексте.

### **semanage**

`semanage command [options]`

Команда «semanage» используется для настройки определенных элементов политики SELinux без необходимости изменения или перекомпиляции политики из исходных текстов. Позволяет настроить сопоставление имен пользователей Linux и пользователей SELinux а также сопоставления контекста безопасности для различных типов объектов, таких как сетевые порты, интерфейсы, узлы (хосты), отображение контекста файлов.

Параметры:



«**command**» - подкоманда. Имеются следующие подкоманды:

- «module» - Управление модулями политики безопасности SELinux
- «import» - Импорт локальных настроек
- «export» - Экспорт локальных настроек
- «login» - Отображение имен пользователей Linux (логинов) в идентификаторы пользователей SELinux.
- «user» - Управление пользователями SELinux (роли и уровни безопасности)
- «port» - Управление типами (перечнями) сетевых портов
- «interface» - Управление типами (перечнями) сетевых интерфейсов
- «node» - Управление типами (перечнями) сетевых узлов
- «fcontext» - Управление определениями контекста безопасности файлов
- «boolean» - Управление «выключателями». Позволяет выборочно включить функциональность

Эти подкоманды будут подробнее разобраны в последующих разделах данного приложения.

«**options**» - параметры подкоманды

Обратите внимание, что команда `semanage login` имеет дело с отображением имен пользователей Linux (логинов) в идентификаторы пользователей SELinux, в то время как команда `semanage user` работает с отображением имен пользователей SELinux в авторизованные наборы ролей. В большинстве случаев администратор должен корректировать только первое сопоставление. Пользователи SELinux в основном определяются базовой политикой и обычно не требует модификации.

## ***semanage module***

```
semanage module [-h] [-n] [-N] [-S STORE] (-a | -r | -e | -d |  
--extract | --list [-C] | --deleteall) [module_name]
```

Команда «`semanage module`» используется для управления модулями политики безопасности SELinux. Некоторые опции:

«**-l**» выводит список модулей, которые в данный момент установлены

«**-a module.pp.bz2**», «**-i module.pp.bz2**», - Устанавливает модуль.

Доступные модули SELinux хранятся в каталоге «`/usr/share/selinux/default/`»

«**-e module**» - Разрешает использование модуля `module`

«**-d module**» - Запрещает использование модуля `module`

«**-r module**» - Удаление модуля `module`

Полную информацию можно получить командой «`man semanage-module`»

## semanage login

```
semanage login [-h] [-n] [-N] [-S STORE] [ --add -s SEUSER -r  
RANGE LOGIN | --delete LOGIN | --deleteall | --extract |  
--list [-C] | --modify -s SEUSER -r RANGE LOGIN ]
```

Команда «semanage login» управляет отображением между пользователем Linux и пользователем SELinux. Например, вы можете определить, что конкретный пользователь или группа пользователей будут входить в систему как пользователь user\_u. Префикс имени группы со знаком «%» указывает название группы. Некоторые опции:

«-l», «--list» - выводит список записей

«-a», «--add» - Добавляет новую запись

«-d», «--delete» - Удаляет запись

«-m», «--modify» - Изменяет запись

«-s SEUser» - Имя пользователя SELinux

«-r RANGE» - Диапазон уровней доступа для MLS/MCS политики безопасности. По умолчанию «s0»

Полную информацию можно получить командой «man semanage-login»

## semanage user

```
semanage user [-h] [-n] [-N] [-S STORE] [ --add ( -L LEVEL -R  
ROLES -r RANGE -s SEUSER selinux_name) | --delete  
selinux_name | --deleteall | --extract | --list [-C] |  
--modify ( -L LEVEL -R ROLES -r RANGE -s SEUSER selinux_name )  
]
```

Команда «semanage user» управляет отображением между пользователем SELinux и ролями и уровнями MLS / MCS. Некоторые опции:

«-l», «--list» - выводит список записей

«-a», «--add» - Добавляет новую запись

«-d», «--delete» - Удаляет запись

«-m», «--modify» - Изменяет запись

«-r RANGE» - Диапазон уровней доступа для MLS/MCS политики безопасности. По умолчанию «s0»

«-L LEVEL», «--level LEVEL» - Задаёт уровень безопасности пользователя по умолчанию «s0»

«-R ROLES», «--roles ROLES» - Задаёт список ролей пользователя SELinux. Роли разделяются пробелами. Список ролей должен быть заключён в кавычки. Так же можно указать несколько опций «-R»

Полную информацию можно получить командой «man semanage-user»

## ***semanage port***

```
semanage port [-h] [-n] [-N] [-S STORE] [ --add -t TYPE -p  
  PROTOCOL -r RANGE port_name | port_range | --delete -p  
  PROTOCOL port_name | port_range | --deleteall | --extract |  
  --list [-C] | --modify -t TYPE -p PROTOCOL -r RANGE port_name  
  port_range ]
```

Команда «semanage port» позволяет создать определение типа порта и включить порт в этот тип. Некоторые опции:

«**-l**», «**--list**» - выводит список записей

«**-a**», «**--add**» - Добавляет новую запись

«**-d**», «**--delete**» - Удаляет запись

«**-m**», «**--modify**» - Изменяет запись

«**-t TYPE**», «**--type TYPE**» - Задаёт имя типа

«**-p PROTO**», «**--proto PROTO**» - Задаёт протокол для указанного порта (tcp | udp)

Полную информацию можно получить командой «man semanage-port»

## ***semanage fcontext***

```
semanage fcontext [-h] [-n] [-N] [-S STORE] [--add ( -t TYPE -f  
  FTYPE -r RANGE -s SEUSER | -e EQUAL ) FILE_SPEC ) | --delete ( -t  
  TYPE -f FTYPE | -e EQUAL ) FILE_SPEC ) | --deleteall |  
  --extract | --list [-C] | --modify ( -t TYPE -f FTYPE -r RANGE  
  -s SEUSER | -e EQUAL ) FILE_SPEC ) ]
```

Команда «semanage fcontext» fcontext используется для управления маркировкой файлов по умолчанию в системе SELinux. Путь к файлам задаётся с помощью регулярного выражения «FILE\_SPEC». Если создаваемый файл удовлетворяет маске, он получает заданную метку безопасности. Некоторые опции:

«**-l**», «**--list**» - выводит список записей

«**-a**», «**--add**» - Добавляет новую запись

«**-d**», «**--delete**» - Удаляет запись

«**-m**», «**--modify**» - Изменяет запись

«**-t TYPE**», «**--type TYPE**» - Задаёт имя типа в метке безопасности

«**-s SEUSER**», «**--seuser SEUSER**» - Задаёт имя пользователя SELinux в метке безопасности.

«**-r RANGE**» - Диапазон уровней доступа для MLS/MCS политики безопасности. По умолчанию «s0»

«**-f [{a,f,d,c,b,s,l,p}]**», «**--ftype [{a,f,d,c,b,s,l,p}]**» - задаёт тип файла (буква соответствует типу файла в выводе команды «ls»). Например, «d» соответствует каталогам, «f» - обычный файл.

Полную информацию можно получить командой «`man semanage-fcontext`»

## ***semanage boolean***

```
semanage boolean [-h] [-n] [-N] [-S STORE] [ --extract |  
--deleteall | --list [-C] | --modify ( --on | --off )  
BOOLEAN_VALUE ]
```

Команда «`semanage boolean`» управляет настройками логических значений в политике SELinux. Логические значения `BOOLEAN_VALUE` указаны в правилах if-then-else, заданных в политике SELinux. Их можно использовать для настройки того, как правила политики SELinux влияют на ограниченный домен.

Некоторые опции:

«**-l**», «**--list**» - выводит список записей

«**-m**», «**--modify**» - Изменяет значение

«**-1**», «**--on**» - Установить значение «Включено»

«**-0**», «**--off**» - Установить значение «Отключено»

Пример:

```
semanage boolean -m --on httpd_can_sendmail
```

Полную информацию можно получить командой «`man semanage-boolean`»

## **Литература**

Вывод команды «`man`»

<https://ru.wikipedia.org/>

<https://wiki.debian.org/>

<https://habr.com/ru/>

<https://defcon.ru/os-security/1264/>

<https://debian-handbook.info/browse/stable/sect.selinux.html>

<https://losst.ru/>

<https://www.opennet.ru/>