

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"

2024

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.

Смена пользователя (*switch user*)

Графическая оболочка не даёт нам входить в систему как суперпользователь (и правильно). Поэтому, чтобы выполнять команды, требующие полномочий суперпользователя мы использовали команду «`su -`». Рассмотрим её подробнее.

Когда требуется сменить пользователя используйте команду

`su ПОЛЬЗОВАТЕЛЬ`

Если «ПОЛЬЗОВАТЕЛЬ» не указан, подразумевается пользователь «root»

Дайте команду «`su`» (без параметров):

```
[user@host-15 ~]$ su
Password:
[root@host-15 ~]#
```

Из текста подсказки видно, что текущим пользователем стал «root».

Но попробуйте дать команду «`fdisk -l`», пользовались на прошлом занятии.

```
[root@host-15 ~]# fdisk -l
bash: fdisk: команда не найдена
```

Дело в том, что без опции «`-l`» команда «`su`» не выполняет специальные сценарии инициализации пользователя «root». Поэтому перечень каталогов в которых «`bash`» ищет

выполняемые файлы остался от пользователя «user».

```
[root@host-15 ~]# echo $PATH  
/home/user/bin:/usr/lib/kf5/bin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/usr/games
```

Права есть, но чтобы дать команду суперпользователя, придётся указывать полный путь к файлу команды, например, «/sbin/fdisk -l»

Для того, чтобы получить полное окружение заданного пользователя используйте команду

```
su -l ПОЛЬЗОВАТЕЛЬ
```

Для смены пользователя на «root» надо дать команду «su -l root» или «su -» (в некоторых версиях linux достаточно просто «su»)

```
[root@host-15 ~]# su -  
[root@host-15 ~]# echo $PATH  
/root/bin:/sbin:/usr/sbin:/usr/local/sbin:/bin:/usr/bin:/usr/local/bin
```

Теперь в списке путей есть «/sbin», «/usr/sbin», «/usr/local/sbin» и все команды суперпользователя доступны без указания пути.

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

Мы давали команду «su» два раза, поэтому и «exit» надо давать дважды.

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 используемом в ЛР облегчённом дистрибутиве её нет.

Для установки дайте команды:

```
su -  
apt-get install sudo
```

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

Дайте команду

```
nano /etc/sudoers
```

Переместите курсор в конец файла. Интерес представляют следующие строки.

```
# root ALL=(ALL) ALL

## Uncomment to allow members of group wheel to execute any command
# WHEEL_USERS ALL=(ALL) ALL

## Same thing without a password
# WHEEL_USERS ALL=(ALL) NOPASSWD: ALL

## Uncomment to allow members of group sudo to execute any command
# SUDO_USERS ALL=(ALL) ALL
```

Во первых, раскомментируйте строку «root ALL=(ALL) ALL». Она даёт права пользователю «root» на выполнение всех команд из под команды «sudo».

Можно добавить строку:

```
user ALL=(ALL) ALL
```

Но удобнее раскомментировать строку «%wheel ALL = (ALL) ALL». Она даёт права на выполнение с помощью «sudo» всех команд всем пользователям группы «wheel». В 3 лабораторной работе мы как раз включили пользователя «user» в эту группу при его создании командой «useradd -m -G **wheel** user».

Строка с «NOPASSWD:» даёт то же но не требует ввода пароля для выполнения команд с помощью «sudo», поэтому не рекомендуется.

При необходимости, вместо последнего «ALL» можно перечислить программы, которые может запускать пользователь или группа. Например, «/usr/bin/apt-get, /bin/mount»

Сохраните файл (Ctrl+O) и выйдите из «nano» (Ctrl+X) и вернитесь в сессию пользователя «user» (командой «exit»).

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 командой «bash».

Теперь, чтобы выполнить команду от имени суперпользователя надо дать команду:

```
sudo КОМАНДА
```

Убедимся, что теперь пользователь «user» может устанавливать недостающие пакеты без смены пользователя:

```
sudo apt-get update
```

```
[user@host-15 ~]$ bash
[user@host-15 ~]$ sudo apt-get update
[sudo] password for user:
Получено: 1 http://ftp.altlinux.org p8/branch/x86_64 release [1091B]
Получено: 2 http://ftp.altlinux.org p8/branch/x86_64-i586 release [537B]
Получено: 3 http://ftp.altlinux.org p8/branch/noarch release [885B]
Получено 2513B за 0s (26,6kB/s).
```

Обратите внимание, что надо вводить не пароль суперпользователя, а пользователя «user», что тоже повышает безопасность.

У «sudo» есть срок (в минутах), в течение которых он не будет просить повторно ввести пароль. Если в файл «/etc/sudoers» добавить строку «Defaults timestamp_timeout=0», то команда «sudo» будет спрашивать пароль при каждом запуске.

Создание пользователей

Смените пользователя на «root».

Создадим пользователя «user1» и зададим ему пароль:

```
useradd -m
passwd user1
<пароль>
<повтор пароля>
```

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@host-15 ~]# ls -la /home/user1
итого 60
drwx----- 8 user1 user1 4096 июн  2 19:34 .
drwxr-xr-x 4 root  root  4096 июн  2 19:34 ..
-rw----- 1 user1 user1  24 фев  7  2015 .bash_logout
-rw----- 1 user1 user1 182 фев  7  2015 .bash_profile
-rw----- 1 user1 user1 124 фев  7  2015 .bashrc
drwx----- 2 user1 user1 4096 фев  7  2015 .cache
drwx----- 4 user1 user1 4096 май 28 19:41 .config
-rw----- 1 user1 user1  20 фев 25  2019 .gtkrc-2.0
drwx----- 3 user1 user1 4096 май 28 17:47 .local
-rw----- 1 user1 user1  17 фев  7  2015 .lpoptions
drwx----- 3 user1 user1 4096 май 28 17:47 .mutt
-rw----- 1 user1 user1 121 фев  7  2015 .rpmmacros
drwx----- 2 user1 user1 4096 май 28 17:47 .ssh
-rwx----- 1 user1 user1 240 фев  7  2015 .xprofile
drwx----- 2 user1 user1 4096 фев  7  2015 .xsession.d
```

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.

Отредактируем файл «.bashrc» пользователя «root»

nano .bashrc

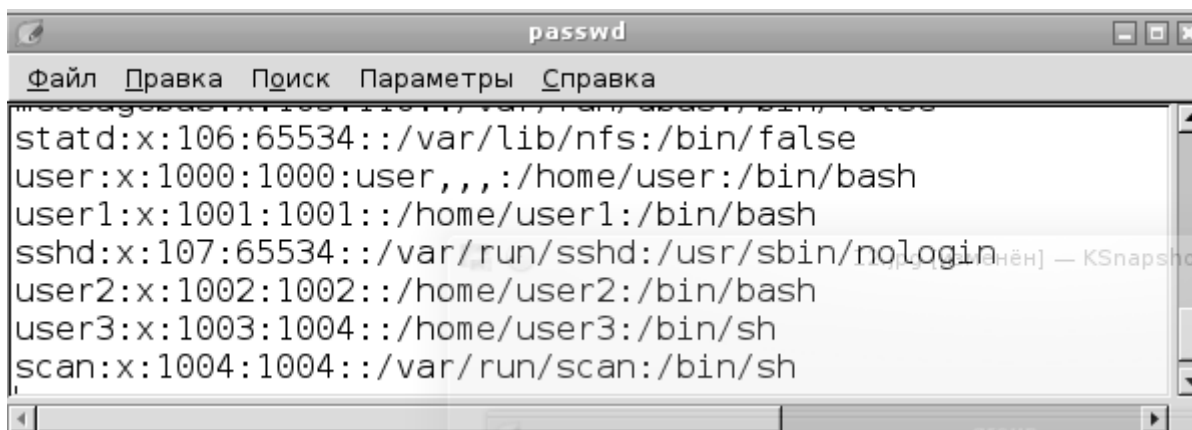
В этот файл файле «.bashrc» пользователь может добавить:

- Алиасы команд с часто употребляемыми опциями. Например, «alias ll='ls -l'». Добавьте эту строку в конец файла
- 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/console/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»:



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

Изменить информацию о пользователе можно прямо в этом файле, 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 отличии 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:

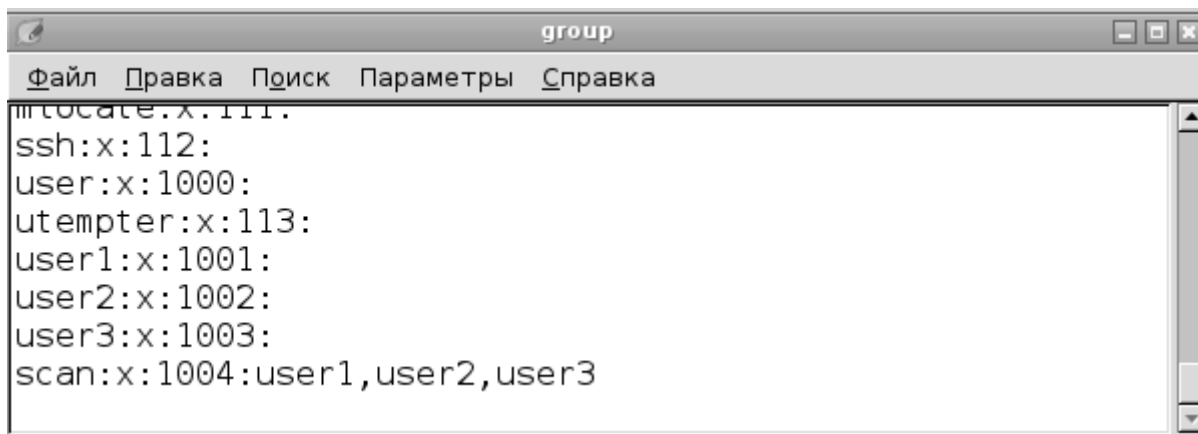
Команда «`usermod -a -G ГРУППА ПОЛЬЗОВАТЕЛЬ`» - добавляет к списку групп заданного пользователя заданную дополнительную группу.

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@host-15:~$ groupadd grp1
root@host-15:~$ usermod -a -G grp1 user1
root@host-15:~$ groups user1
user1 : user1 grp1
root@host-15:~$ groups user
user : user wheel
```

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



```
group
Файл Правка Поиск Параметры Справка
итого: x:111:
ssh:x:112:
user:x:1000:
utempter:x:113:
user1:x:1001:
user2:x:1002:
user3:x:1003:
scan:x:1004:user1,user2,user3
```

Этот файл можно редактировать 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.

Просмотр существующих прав

Рассмотрим пример прав доступа:

```
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 «-»

Приведу описание этих прав:

Категории пользователей	Символ	Для файла	Для каталога
Для всех категорий	r	Чтение файла	Просмотр списка файлов каталога
Для всех категорий	w	Запись в файл	Создание и удаление файлов и подкаталогов.
Для всех категорий	x	Права на выполнение	Права на доступ в каталог

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

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 «х», 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 Применяется для системных утилит.	Не применяется
Права группы	s	Любой пользователь может запустить файл с правами группы Применяется для системных утилит.	Все файлы, создаваемые в каталоге принадлежат группе, владеющей каталогом 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 Применяется для каталогов временных файлов

Изменение прав на файлы и каталоги

Разберём права на домашние каталоги пользователей полученные при создании:

```
root@host-15:~$ ls -l /home
итого 8
drwx----- 12 user  user  4096 июн  2 19:33 user
drwx-----  8 user1 user1  4096 июн  2 19:44 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).

У группы и прочих пользователей, (вторая т третья тройка «- - -») - для безопасности, отлично. 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 и каталоги для обмена данными в каталоге «/usr/share»

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 root. 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 770 /home/user1/testdir` — даёт все права для пользователя и группы.

`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 подкаталогов. 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@host-15:~$ mkdir /home/user1/work
root@host-15:~$ ls -l /home/user1
итого 4
drwxr-xr-x 2 root root 4096 июн  2 20:56 work
root@host-15:~$ chown user1:user1 /home/user1/work/
root@host-15:~$ ls -l /home/user1
итого 4
drwxr-xr-x 2 user1 user1 4096 июн  2 20:56 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.

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.

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

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

```

root@host-15:~$ touch file
root@host-15:~$ 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@host-15:~$ setfacl -m "u:user1:rwX" file
root@host-15:~$ setfacl -m "g:grp1:r-x" file
root@host-15:~$ 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

Этот раздел для ознакомления. Выполнять не надо.

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.

К сожалению, ALT Linux надстройку SELinux поддерживает только в сертифицированной версии Альт Линукс СПТ.

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 на чтение и запись имеет 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

Пользователи не должны иметь доступ к файлам других пользователей. Не забудьте дать права сервису распознавания. Выберите для «`in`» и «`out`» подходящий родительский каталог и не забудьте дать права на доступ в него, при необходимости.

- Создать каталог «`DSP`», в который будут выкладывать файлы пользователи группы «`dsp`» для обмена между собой. 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, а сервис права на запись.

Контрольные вопросы

В этом разделе приведены типичные вопросы, задаваемые на защите лабораторной работы. Ответы в отчёт включать не надо.

- Почему в традиционной системе защиты рекомендуется, чтобы сервисы не работали от имени пользователя «`root`»? Не пишите просто «Для безопасности», обоснуйте.
- Зачем нужна опция «`-l`» в команде «`su`»?
- Как дать пользователю права на выполнение команд через «`sudo`»?
- Зачем в команде «`useradd`» опция «`-m`»?
- Какой командой можно сменить пароль пользователя?
- В каком файле хранится информация о пользователях?

- В каких группах окажется пользователь после команд «usermod -G group1 user1; usermod -G group2 user1»?
- Какой командой можно получить список групп, в которые входит пользователь пользователь «user»?
- Посмотрите строку вывода команды «ls -l /home» - «d r w x - - - - -
27 user user 4096 май 19 2023 user». Какие пользователи имеют доступ в каталог «user».
- Какую команду надо дать, чтобы на создаваемые файлы имели права на чтение и запись пользователь-владелец и группа-владелец?

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.

Структура файла /etc/passwd

Файл /etc/passwd — это текстовый файл, описывающий учетные записи пользователей для входа в систему. Он должен иметь разрешение на чтение для всех пользователей (многие утилиты, такие как ls(1), используют его для сопоставления идентификаторов пользователей с именами пользователей), но доступ на запись только для суперпользователя.

В настоящее время, зашифрованные пароли хранятся не в этом файле, а в файле /etc/shadow, который доступен для чтения только суперпользователю.

Каждая строка файла описывает одного пользователя и содержит семь полей, разделенных двоеточиями:

name:password:UID:GID:GECOS:directory:shell

«name» - Это имя пользователя для входа. Оно не должно содержать заглавных букв.

«password» - Раньше здесь был зашифрованный пароль. Теперь буква «x».

«UID» - Идентификатор пользователя. 0 зарезервировано для пользователя «root»

«GID» - Идентификатор первичной группы данного пользователя. Дополнительные группы для пользователя хранятся в файле «/etc/group»

«GECOS» - Это поле (иногда называемое «полем комментария») является необязательным и используется только в информационных целях. Обычно он содержит полное имя пользователя.

«directory» - Это домашний каталог пользователя: начальный каталог, в который

попадает пользователь после входа в систему. Значение в этом поле используется для установки переменной среды HOME.

«shell» - Это программа, запускаемая при входе в систему (если она пуста, используйте /bin/sh). Если установлено значение несуществующего исполняемого файла, пользователь не сможет войти в систему через login. Значение в этом поле используется для установки переменной среды SHELL.

sudo

```
sudo -h | -K | -k | -V
sudo -v [-ABknS] [-g group] [-h host] [-p prompt] [-u user]
sudo -l [-ABknS] [-g group] [-h host] [-p prompt] [-U user] [-u
user] [command]
sudo [-ABbEHnPS] [-C num] [-g group] [-h host] [-p prompt] [-T
timeout] [-u user] [VAR=value] [-i | -s] [command]
```

sudo позволяет пользователю выполнять команду от имени суперпользователя или другого пользователя, как указано в политике безопасности.

«sudo» поддерживает архитектуру плагинов для политик безопасности и ведения журнала ввода/вывода. Третьи стороны могут разрабатывать и распространять свои собственные плагины политик и журналов ввода-вывода для бесперебойной работы с интерфейсом «sudo». Политика безопасности по умолчанию — «sudoers», которая настраивается через файл «/etc/sudoers» или через LDAP.

Политика безопасности определяет, какие привилегии (если таковые имеются) есть у пользователя. Политика может требовать, чтобы пользователи аутентифицировали себя с помощью пароля или другого механизма аутентификации. Если требуется аутентификация, sudo завершится, если пароль пользователя не будет введен в течение настраиваемого срока. Это ограничение зависит от политики; время ожидания запроса пароля по умолчанию для политики безопасности «sudoers» составляет 5 минут.

Политики безопасности могут поддерживать кэширование учетных данных, чтобы позволить пользователю снова запустить «sudo» в течение определенного периода времени без необходимости аутентификации. Политика «sudoers» кэширует учетные данные в течение 5 минут, если она не переопределена в sudoers(5). Запустив «sudo» с опцией «-v», пользователь может обновить кэшированные учетные данные.

Политики безопасности могут регистрировать успешные и неудачные попытки использования «sudo». Если настроен плагин ввода-вывода, ввод и вывод выполняемой команды также могут регистрироваться.

Параметры:

«-A, --askpass» - Обычно, если sudo требует пароль, он считывает его с

терминала пользователя. Если указана опция `-A (askpass)`, выполняется (возможно, графическая) вспомогательная программа для чтения пароля пользователя и вывода пароля на стандартный вывод. Если установлена переменная среды `SUDO_ASKPASS`, она указывает путь к вспомогательной программе. В противном случае, если `sudo.conf(5)` содержит строку, определяющую программу `Askpass`, будет использоваться это значение.

«`-B, --bell`» - Дать звуковой сигнал при запросе пароля при наличии терминала. Эта опция не имеет эффекта, если используется программа `Askpass`.

«`-b, --background`» - Запускает заданную команду в фоновом режиме. Обратите внимание, что невозможно использовать управление заданиями оболочки для управления фоновыми процессами, запускаемыми `sudo`. Большинство интерактивных команд не будут работать должным образом в фоновом режиме.

«`-C num, --close-from=num`» - Закрывает все файловые дескрипторы, большие или равные `num`, перед выполнением команды. Значения меньше трех не допускаются. По умолчанию `sudo` при выполнении команды закроет все открытые файловые дескрипторы, кроме стандартного ввода, стандартного вывода и стандартных ошибок. Политика безопасности может ограничить возможность пользователя использовать эту опцию. Политика `sudoers` разрешает использование опции `-C` только в том случае, если администратор включил опцию `closefrom_override`.

«`-E, --preserve-env`» - Указывает политике безопасности, что пользователь желает сохранить существующие переменные среды. Политика безопасности может вернуть ошибку, если у пользователя нет разрешения на сохранение среды.

«`--preserve-env=list`» - Указывает политике безопасности, что пользователь желает добавить список переменных среды, разделенных запятыми, к переменным среды, сохраненным из среды пользователя. Политика безопасности может вернуть ошибку, если у пользователя нет разрешения на сохранение среды.

«`-e, --edit`» - Редактирует один или несколько файлов вместо запуска команды. Вместо имени пути при просмотре политики безопасности используется строка «`sudoedit`». Если пользователь авторизован политикой, выполняются следующие шаги:

1. Из файлов, подлежащих редактированию, создаются временные копии, владельцем которых является вызвавший пользователь.

2. Для редактирования временных файлов запускается редактор, указанный политикой. Политика `sudoers` использует переменные среды `SUDO_EDITOR`, `VISUAL` и `EDITOR` (в указанном порядке). Если ни один из `SUDO_EDITOR`, `VISUAL` или `EDITOR` не установлен, используется первая программа, указанная в опции редактора `sudoers(5)`.

3. Если они были изменены, временные файлы копируются обратно в исходное

местоположение, а временные версии удаляются. Чтобы предотвратить редактирование несанкционированных файлов, применяются следующие ограничения, если это явно не разрешено политикой безопасности:

- Символические ссылки не редактируются (версия 1.8.15 и выше).
- Символические ссылки на редактируемом пути не выполняются, если родительский каталог доступен для записи вызывающему пользователю, если только этот пользователь не является пользователем root (версия 1.8.16 и выше).
- Файлы, расположенные в каталоге, доступном для записи вызывающему пользователю, не могут быть отредактированы, если только этот пользователь не является пользователем root (версия 1.8.16 и выше).
- Пользователям никогда не разрешается редактировать специальные файлы устройства.

Если указанный файл не существует, он будет создан. Обратите внимание, что в отличие от большинства команд, запускаемых sudo, редактор запускается со средой вызывающего пользователя. Если по какой-либо причине sudo не сможет обновить файл отредактированной версией, пользователь получит предупреждение, а отредактированная копия останется во временном файле.

«-g group, --group=group» - Выполняет команду, указав в качестве основной группы значение group вместо основной группы, указанной в записи базы данных паролей целевого пользователя. Группа может представлять собой либо имя группы, либо числовой идентификатор группы (GID) с префиксом символа «#» (например, #0 для GID 0). При запуске команды в качестве GID многие оболочки требуют, чтобы символ «#» был экранирован обратной косой чертой «\». Если опция «-u» не указана, команда будет запущена от имени вызвавшего пользователя. В любом случае основная группа будет установлена как «group». Политика «sudoers» позволяет указывать любую группу целевых пользователей с помощью параметра «-g», если параметр «-P» не используется.

«-H, --set-home» - Запрашивает, чтобы политика безопасности установила переменную среды HOME в домашний каталог, указанный в записи базы данных паролей целевого пользователя. В зависимости от политики это может быть поведением по умолчанию.

«-h, --help» - Вывести короткое справочное сообщение на стандартный вывод и выйти.

«-h host, --host=host» - Запускает команду на указанном хосте, если плагин политики безопасности поддерживает удаленные команды. Обратите внимание, что плагин «sudoers» в настоящее время не поддерживает запуск удаленных команд. Можно

использовать в сочетании с опцией «-l» для вывода списка привилегий пользователя для удаленного хоста.

«-i, --login» - Запускает оболочку, указанную в записи базы данных паролей целевого пользователя, в качестве оболочки входа в систему. Это означает, что файлы ресурсов, специфичные для входа в систему, такие как .profile, .bash_profile или .login, будут считываться оболочкой. Если указана команда, она передается оболочке для выполнения через опцию оболочки «-c». Если команда не указана, выполняется интерактивная оболочка. sudo пытается перейти в домашний каталог этого пользователя перед запуском оболочки. Команда запускается в среде, аналогичной той, которую пользователь получает при входе в систему. Обратите внимание, что большинство оболочек ведут себя по-разному, когда указана команда, по сравнению с интерактивным сеансом; за подробностями обратитесь к руководству по оболочке. В разделе «Среда команд» в руководстве sudoers(5) описано, как опция «-i» влияет на среду, в которой выполняется команда, когда используется политика sudoers.

«-K, --remove-timestamp» - Аналогичен параметру -k, за исключением того, что он полностью удаляет кэшированные учетные данные пользователя и не может использоваться вместе с командой или другим параметром. Эта опция не требует пароля. Не все политики безопасности поддерживают кэширование учетных данных.

«-k, --reset-timestamp» - При использовании без команды аннулирует кэшированные учетные данные пользователя. Другими словами, при следующем запуске sudo потребуются пароль. Эта опция не требует пароля и была добавлена, чтобы позволить пользователю отзываться разрешения sudo из файла .logout. При использовании в сочетании с командой или опцией, для которой может потребоваться пароль, эта опция приведет к тому, что sudo будет игнорировать кэшированные учетные данные пользователя. В результате sudo запросит пароль (если он требуется политикой безопасности) и не будет обновлять кэшированные учетные данные пользователя. Не все политики безопасности поддерживают кэширование учетных данных.

«-l, --list» - Если команда не указана, вывести список разрешенных (и запрещенных) команд для вызывающего пользователя (или пользователя, указанного опцией -U) на текущем хосте. Более длинный формат списка используется, если этот параметр указан несколько раз и политика безопасности поддерживает подробный формат вывода. Если команда указана и разрешена политикой безопасности, полный путь к команде отображается вместе со всеми аргументами командной строки. Если команда указана, но не разрешена политикой, sudo завершится со значением статуса 1.

«-n, --non-interactive» - Не запрашивает у пользователя какой-либо ввод

данных. Если для запуска команды требуется пароль, `sudo` отобразит сообщение об ошибке и завершит работу.

«-P, --preserve-groups» - Сохраняет неизменным вектор группы вызывающего пользователя. По умолчанию политика `sudoers` инициализирует вектор группы списком групп, членом которых является целевой пользователь. Однако реальные и действующие идентификаторы групп по-прежнему соответствуют целевому пользователю.

«-p prompt, --prompt=prompt» - Использует собственный запрос пароля с дополнительными `escape`-последовательностями. Политикой «`sudoers`» поддерживаются следующие `escape`-последовательности:

- %H расширяется до имени хоста, включая имя домена (включено, если имя хоста компьютера полностью задано или в `sudoers(5)` установлен параметр `fqdn`)
- %h расширяется до имени локального хоста без имени домена
- %p расширяется до имени пользователя, чей пароль запрашивается (уважает флаги `rootpw`, `targetpw` и `runaspw` в `sudoers(5)`)
- %U расширяется до имени пользователя, от имени которого будет выполняться команда (по умолчанию `root`, если не указана опция `-u`) %u расширяется до имени входа вызывающего пользователя
- %% два последовательных символа «%» сворачиваются в один символ «%»

Пользовательский запрос переопределит запрос по умолчанию, указанный либо в политике безопасности, либо в переменной среды `SUDO_PROMPT`. В системах, использующих `PAM`, пользовательское приглашение также будет переопределять приглашение, указанное модулем `PAM`, если в `sudoers` не отключен флаг `passprompt_override`.

«-S, --stdin» - Записывает подсказку в стандартный поток ошибок и прочитает пароль со стандартного потока ввода вместо использования терминального устройства.

«-s, --shell» - Запускает оболочку, указанную в переменной среды `SHELL`, если она установлена, или оболочку, указанную в записи базы данных паролей вызывающего пользователя. Если указана команда, она передается оболочке для выполнения через опцию оболочки «-c». Если команда не указана, выполняется интерактивная оболочка. Обратите внимание, что большинство оболочек ведут себя по-разному, когда указана команда, по сравнению с интерактивным сеансом; за подробностями обратитесь к руководству по оболочке.

«-U user, --other-user=user» - Используется в сочетании с опцией `-l` для вывода списка привилегий пользователя, а не вызвавшего пользователя. Политика безопасности может ограничивать перечисление привилегий других пользователей. Политика `sudoers` позволяет использовать эту опцию только пользователю `root` или

пользователю с привилегиями ALL на текущем хосте.

«-V, --version» - Печатает строку версии sudo, а также строку версии плагина политики безопасности и всех плагинов ввода-вывода. Если вызывающий пользователь уже является пользователем root, опция -V отобразит аргументы, переданные для настройки при сборке sudo, а плагины могут отображать более подробную информацию, например параметры по умолчанию.

«-v, --validate» - Обновляет кэшированные учетные данные пользователя, при необходимости проверив его подлинность. Для плагина sudoers это продлевает тайм-аут sudo еще на 5 минут по умолчанию, но не запускает команду. Не все политики безопасности поддерживают кэширование учетных данных.

«--» - указывает, что sudo следует прекратить обработку аргументов командной строки

Переменные среды, которые необходимо установить для команды, также можно передавать в командной строке в форме VAR=значение, например, LD_LIBRARY_PATH=/usr/local/pkg/lib. На переменные, передаваемые в командной строке, распространяются ограничения, налагаемые плагином политики безопасности. Политика sudoers подвергает переменные, передаваемые в командной строке, тем же ограничениям, что и обычные переменные среды, за одним важным исключением. Если в sudoers установлена опция setenv, у запускаемой команды установлен тег SETENV или соответствует команда ALL, пользователь может устанавливать переменные, которые в противном случае были бы запрещены. Дополнительную информацию смотрите в sudoers(5).

useradd

```
useradd [options] LOGIN
useradd -D
useradd -D [options]
```

Команда «useradd» позволяет создать нового пользователя.

При вызове без опции -D команда «useradd» создает новую учетную запись пользователя, используя значения, указанные в командной строке, а также значения по умолчанию. В зависимости от параметров командной строки команда обновляет системные файлы, а также может создать домашний каталог нового пользователя и скопировать исходные файлы. По умолчанию для нового пользователя также будет создана группа (см. -g, -N, -U и USERGROUPS_ENAB).

Параметры:

«-b, --base-dir BASE_DIR» - Базовый каталог системы по умолчанию, если не указан параметр -d HOME_DIR. BASE_DIR объединяется с именем учетной записи для

определения домашнего каталога. Если опция `-m` не используется, `BASE_DIR` должен существовать. Если этот параметр не указан, «useradd» будет использовать базовый каталог, указанный переменной `HOME` в файле «/etc/default/useradd», или «/home» по умолчанию.

«-c, --comment COMMENT» - Любая текстовая строка. Обычно это краткое описание входа в систему, которое в настоящее время используется в качестве поля для полного имени пользователя.

«-d, --home-dir HOME_DIR» - Новый пользователь будет создан с использованием `HOME_DIR` в качестве домашнего каталога пользователя. По умолчанию логин пользователя добавляется к `BASE_DIR` и используется в качестве имени каталога. Каталог `HOME_DIR` не обязательно должен существовать, но не будет создан, если он отсутствует.

«-D, --defaults» - См. ниже подраздел «Изменение значений по умолчанию»

«-e, --expiredate EXPIRE_DATE» - Дата, когда учетная запись пользователя будет отключена. Дата указывается в формате ГГГГ-ММ-ДД. Если этот параметр не указан, будет использована дата истечения срока действия по умолчанию, указанная переменной `EXPIRE` в файле «/etc/default/useradd», или пустую строку (без срока действия) по умолчанию.

«-f, --inactive INACTIVE» - Количество дней после истечения срока действия пароля до момента окончательного отключения учетной записи. Значение 0 отключает учетную запись, как только истечет срок действия пароля, а значение -1 отключает эту функцию. Если не указано, «useradd» будет использовать период бездействия по умолчанию, указанный переменной `INACTIVE` в файле «/etc/default/useradd», или -1 по умолчанию.

«-g, --gid GROUP» - Имя или номер первичной группы пользователя. Имя или номер группы должны существовать. Если не указано иное, поведение «useradd» будет зависеть от переменной `USERGROUPS_ENAB` в файле «/etc/login.defs». Если для этой переменной установлено значение «yes» (или в командной строке указана опция `-U/--user-group`), для пользователя будет создана группа с тем же именем, что и ее логин. Если для переменной установлено значение no (или в командной строке указано `-N/--no-user-group`), «useradd» установит для основной группы нового пользователя значение, указанное в `GROUP`. переменная в файле «/etc/default/useradd» или 100 по умолчанию.

«-G, --groups GROUP1[,GROUP2,...[,GROUPN]]» - Список дополнительных групп, членом которых также является пользователь. Каждая группа

отделяется от следующей запятой без промежуточных пробелов. На группы распространяются те же ограничения, что и на группу, заданную опцией «-g». По умолчанию пользователь принадлежит только первичной группе.

«-h, --help» - Отобразить краткую справку по команде и выйти.

«-k, --skel SKEL_DIR» - Скелетный каталог, содержащий файлы и каталоги, которые необходимо скопировать в домашний каталог пользователя, когда домашний каталог создается с помощью данной команды. Этот параметр действителен только в том случае, если указан параметр -m (или --create-home). Если этот параметр не установлен, каталог скелета определяется переменной SKEL в файле «/etc/default/useradd» или, по умолчанию, «/etc/skel». Если возможно, копируются списки ACL и расширенные атрибуты.

«-K, --key KEY=VALUE» - Переопределяет значения по умолчанию в файле «/etc/login.defs» (UID_MIN, UID_MAX, UMASK, PASS_MAX_DAYS и другие). Например: -K PASS_MAX_DAYS=-1 можно использовать при создании системной учетной записи, чтобы отключить устаревание пароля, даже если системная учетная запись вообще не имеет пароля. Можно указать несколько опций -K, например: -K UID_MIN=100. -K UID_MAX=499

«-l, --no-log-init» - Не добавлять пользователя в базы данных «lastlog» и «faillog». По умолчанию записи пользователя добавляются в базы данных «lastlog» и «faillog», чтобы избежать повторного использования записи ранее удаленного пользователя.

«-m, --create-home» - Создавать домашний каталог пользователя, если он не существует. Файлы и каталоги, содержащиеся в скелетном каталоге (который можно определить с помощью опции -k), будут скопированы в домашний каталог. По умолчанию, если этот параметр не указан и CREATE_HOME не включен, домашние каталоги не создаются.

«-M» - Не создавать домашний каталог пользователя, даже если для параметра «CREATE_HOME» в файле «/etc/login.defs» установлено значение «yes».

«-N, --no-user-group» - Не создавать группу с тем же именем, что и у пользователя, а добавьте пользователя в группу, указанную опцией -g или переменной GROUP в файле «/etc/default/useradd». Поведение по умолчанию (если параметры -g, -N и -U не указаны) определяется переменной USERGROUPS_ENAB в файле «/etc/login.defs».

«-o, --non-unique» - Разрешить создание учетной записи пользователя с

повторяющимся (неуникальным) UID. Эта опция действительна только в сочетании с «-u».

«-p, --password PASSWORD» - Зашифрованный пароль, возвращаемый `crypt(3)`. По умолчанию пароль отключен. Примечание. Этот вариант не рекомендуется использовать, поскольку пароль (или зашифрованный пароль) будет виден пользователям, просматривающим список процессов. Вы должны убедиться, что пароль соответствует политике паролей системы.

«-r, --system» - Создает учетную запись системного пользователя. Пользователи будут созданы без информации об устаревании в «/etc/shadow», а их числовые идентификаторы выбираются в диапазоне `SYS_UID_MIN-SYS_UID_MAX`, определенном в «/etc/login.defs», вместо `UID_MIN-UID_MAX` (и их аналогов `GID` для создания групп). Обратите внимание, что «useradd» не создаст домашний каталог для такого пользователя, независимо от настройки `CREATE_HOME` в файле «/etc/login.defs». Вам необходимо указать опцию -m, если вы хотите создать домашний каталог для системной учетной записи.

«-R, --root CHROOT_DIR» - команда использует файлы конфигурации из каталога `CHROOT_DIR`.

«-s, --shell SHELL» - Имя интерпретатора команд (оболочки) для пользователя. По умолчанию это поле остается пустым, что заставляет систему выбирать оболочку входа по умолчанию, указанную переменной `SHELL` в «/etc/default/useradd», или пустую строку по умолчанию.

«-u, --uid UID» - Числовое значение идентификатора пользователя. Это значение должно быть уникальным, если не используется опция -o. Значение должно быть неотрицательным. По умолчанию используется наименьшее значение идентификатора, большее или равное `UID_MIN`, и больше, чем у любого другого пользователя. См. также параметр «-r».

«-U, --user-group» - Создает группу с тем же именем, что и у пользователя, и добавляет пользователя в эту группу. Поведение по умолчанию (если параметры -g, -N и -U не указаны) определяется переменной `USERGROUPS_ENAB` в «/etc/login.defs».

«-Z, --selinux-user SEUSER» - Пользователь SELinux, соответствующий данному пользователю. По умолчанию это поле остается пустым, в результате чего система выбирает пользователя SELinux по умолчанию.

usermod

`usermod [options] LOGIN`

Изменяет учетную запись пользователя.

Команда «usermod» изменяет информацию о пользователе в системных файлах.

Параметры:

«-a, --append» - Добавьте пользователя в дополнительные группы. Используйте только с опцией -G

«-c, --comment COMMENT» - Новое значение поля комментария к файлу паролей пользователей.

«-d, --home HOME_DIR» - Новый домашний каталог пользователя. Если указана опция -m, содержимое текущего домашнего каталога будет перемещено в новый домашний каталог, который создается, если он еще не существует.

«-e, --expiredate EXPIRE_DATE» - Дата, когда учетная запись пользователя будет отключена. Дата указывается в формате ГГГГ-ММ-ДД. Пустой аргумент EXPIRE_DATE отключит истечение срока действия учетной записи. Для этой опции требуется файл «/etc/shadow». Запись в «/etc/shadow» будет создана, если ее не было.

«-f, --inactive INACTIVE» - Количество дней после истечения срока действия пароля до момента окончательного отключения учетной записи. Значение 0 отключает учетную запись, как только истечет срок действия пароля, а значение -1 отключает эту функцию. Для этой опции требуется файл /etc/shadow. Запись в /etc/shadow будет создана, если ее не было.

«-g, --gid GROUP» - Имя или номер первичной группы пользователя. Группа должна существовать. Любой файл из домашнего каталога пользователя, принадлежащий предыдущей первичной группе пользователя, будет принадлежать этой новой группе. Групповое владение файлами за пределами домашнего каталога пользователя должно быть исправлено вручную.

«-G, --groups GROUP1[,GROUP2,...[,GROUPN]]» - Список дополнительных групп, членом которых также является пользователь. Каждая группа отделяется от следующей запятой без промежуточных пробелов. На группы распространяются те же ограничения, что и на группу, указанную с помощью опция -g. Если пользователь в настоящее время является членом группы, которой нет в списке, он будет удален из группы. Это поведение можно изменить с помощью опции -a, которая добавляет пользователя в текущий список дополнительных групп.

«-l, --login NEW_LOGIN» - Имя пользователя будет изменено с LOGIN на NEW_LOGIN. Больше ничего не изменилось. В частности, домашний каталог или почтовый ящик пользователя, вероятно, следует переименовать вручную, чтобы отразить новое имя.

«-L, --lock» - Блокировка пароля пользователя. Ставит знак '!' перед зашифрованным паролем, что фактически отключает пароль. Эту опцию нельзя использовать

с -р или -U. Примечание. Если вы хотите заблокировать учетную запись (а не только доступ с помощью пароля), вам также следует установить EXPIRE_DATE равным 1.

«-m, --move-home» - Перемещает содержимое домашнего каталога пользователя в новое место. Этот параметр действителен только в сочетании с параметром -d (или --home). usermod попытается адаптировать владельца файлов и скопировать режимы, ACL и расширенные атрибуты, но впоследствии могут потребоваться изменения вручную.

«-o, --non-unique - При использовании с опцией -u этот параметр позволяет изменить идентификатор пользователя на неуникальное значение.

«-p, --password PASSWORD» - Зашифрованный пароль, возвращаемый crypt(3). Примечание. Этот вариант не рекомендуется использовать, поскольку пароль (или зашифрованный пароль) будет виден пользователям, просматривающим список процессов. Пароль будет записан в локальном файле /etc/passwd или /etc/shadow. Это может отличаться от базы данных паролей, настроенной в вашей конфигурации PAM. Вы должны убедиться, что пароль соответствует политике паролей системы.

«-R, --root CHROOT_DIR» - Использует файлы конфигурации из каталога CHROOT_DIR.

«-s, --shell SHELL» - Имя новой оболочки входа пользователя. Если задать пустое значение, система выберет оболочку входа по умолчанию.

«-u, --uid UID» - Новое числовое значение идентификатора пользователя. Это значение должно быть уникальным, если не используется опция -o. Значение должно быть неотрицательным. Почтовый ящик пользователя и любые файлы, принадлежащие пользователю и расположенные в домашнем каталоге пользователя, будут иметь идентификатор пользователя файла, измененный автоматически. Право собственности на файлы за пределами домашнего каталога пользователя необходимо установить вручную. Никакие проверки не будут выполняться в отношении UID_MIN, UID_MAX, SYS_UID_MIN или SYS_UID_MAX из /etc/login.defs.

« -v, --add-subuids FIRST-LAST» - Добавляет в учетную запись пользователя ряд подчиненных идентификаторов. Эту опцию можно указать несколько раз, чтобы добавить несколько диапазонов к учетной записи пользователя. Никакие проверки не будут выполняться в отношении SUB_UID_MIN, SUB_UID_MAX или SUB_UID_COUNT из /etc/login.defs.

« -V, --del-subuids FIRST-LAST» - Удаляет ряд подчиненных идентификаторов из учетной записи пользователя. Эту опцию можно указать несколько раз, чтобы удалить несколько диапазонов для учетной записи пользователя. Если указаны оба — del-subuids и —add-subuids, удаление всех подчиненных диапазонов uid происходит до

добавления какого-либо подчиненного диапазона uid. Никакие проверки не будут выполняться в отношении SUB_UID_MIN, SUB_UID_MAX или SUB_UID_COUNT из /etc/login.defs.

«-Z, --selinux-user SEUSER» - Новый пользователь SELinux для входа в систему. Пустой SEUSER удалит сопоставление пользователя SELinux для пользователя LOGIN (если таковое имеется).

userdel

userdel [options] LOGIN

Удалить учетную запись пользователя и связанные с ней файлы

Команда userdel изменяет системные файлы, удаляя все записи, относящиеся к имени пользователя LOGIN. Именованный пользователь должен существовать.

Параметры:

«-f, --force» - Этот параметр принудительно удаляет учетную запись пользователя, даже если пользователь все еще вошел в систему. Он также заставляет userdel удалить домашний каталог и почтовую очередь пользователя, даже если другой пользователь использует тот же домашний каталог или если почтовая очередь не принадлежит указанному пользователю. Если для USERGROUPS_ENAB в /etc/login.defs задано значение «да» и существует группа с тем же именем, что и у удаленного пользователя, то эта группа будет удалена, даже если она все еще основная группа другого пользователя. Примечание. Этот параметр опасен и может привести вашу систему в несогласованное состояние.

«-h, --help» - Отобразить краткую справку по команде и выйти.

«-r, --remove» - Файлы в домашнем каталоге пользователя будут удалены вместе с самим домашним каталогом и почтовой очередью пользователя. Файлы, расположенные в других файловых системах, придется искать и удалять вручную.

«-R, --root CHROOT_DIR» - Использует файлы конфигурации из каталога CHROOT_DIR.

«-Z, --selinux-user» - Удалите все сопоставления пользователей SELinux для заданного пользователя.

groupadd

groupadd [options] group

Создать новую группу.

Команда groupadd создает новую учетную запись группы, используя значения, указанные в командной строке, а также значения по умолчанию из системы. Новая группа будет внесена в системные файлы по мере необходимости.

Параметры:

« -f, --force» - Эта опция приводит к тому, что команда просто завершает работу со статусом успеха, если указанная группа уже существует. При использовании с -g и указанным GID уже существует, выбирается другой (уникальный) GID (т. е. -g отключается).

« -g, --gid GID» - Числовое значение идентификатора группы. Это значение должно быть уникальным, если не используется опция -o. Значение должно быть неотрицательным. По умолчанию используется наименьшее значение идентификатора, большее или равное GID_MIN и больше, чем любая другая группа. См. также параметр -г и описание GID_MAX.

«-h, --help» - Отобразить краткую справку по команде и выйти.

«-K, --key KEY=VALUE» - Переопределяет значения по умолчанию в файле /etc/login.defs (GID_MIN, GID_MAX и другие). Можно указать несколько опций -K. Пример: -K GID_MIN=100 -K GID_MAX=499 Примечание. -K GID_MIN=10,GID_MAX=499 пока не работает.

«-o, --non-unique» - Эта опция позволяет добавить группу с неуникальным GID.

«-p, --password PASSWORD» - Зашифрованный пароль, возвращаемый crypt(3). По умолчанию пароль отключен. Примечание. Этот вариант не рекомендуется использовать, поскольку пароль (или зашифрованный пароль) будет виден пользователям, перечисляющим процессы. Вы должны убедиться, что пароль соответствует политике паролей системы.

«-r, --system» - Создает системную группу. Числовые идентификаторы новых системных групп выбираются в диапазоне SYS_GID_MIN-SYS_GID_MAX, определенном в файле login.defs, вместо GID_MIN-GID_MAX.

«-R, --root CHROOT_DIR» - Использует файлы конфигурации из каталога CHROOT_DIR.«xxx» -

chown

```
chown [OPTION]... [OWNER] [:[GROUP]] FILE...
```

```
chown [OPTION]... --reference=RFILE FILE...
```

Изменяет владельца и группу файла или каталога.

chown изменяет права пользователя и/или группы на каждый данный файл. Если указан только владелец (имя пользователя или числовой идентификатор пользователя), этот пользователь становится владельцем заданных файлов, а группа файлов не изменяется. Если за владельцем следует двоеточие и имя группы (или числовой идентификатор группы) без пробелов между ними, группа-владелец файлов также меняется. Если указано двоеточие, но нет имени группы за именем пользователя, этот пользователь становится владельцем файлов, а группа файлов меняется на первичную группу этого пользователя. Если указаны двоеточие

и группа, но владелец опущен, будет изменена только группа. В этом случае `chown` выполняет ту же функцию, что и `chgrp`. Если указано только двоеточие или весь операнд пуст, ни владелец, ни группа не изменяются.

Параметры:

«-c, --changes» - подобно «-v», но выводит сообщения только при внесении изменений

«-f, --silent, --quiet» - подавляет большинство сообщений об ошибках

«-v, --verbose» - выводить диагностику для каждого обработанного файла

«--dereference» - изменяет владельца файла на который ссылается символическая ссылка (это значение по умолчанию), а не самой ссылки

«-h, --no-dereference» - изменять сами символические ссылки вместо ссылочного файла (полезно только в системах, которые могут изменить владельца символической ссылки)

«--from=CURRENT_OWNER:CURRENT_GROUP» - менять владельца и/или группу каждого файла только в том случае, если его текущий владелец и/или группа соответствуют указанным здесь. Любой из них может быть опущен, и в этом случае совпадение для опущенного атрибута не требуется.

«--no-preserve-root» - не обрабатывать корневой каталог (по умолчанию)

«--preserve-root» - Ошибка при рекурсивной обработке корневого каталога.

«--reference=RFILE» - использовать владельца и группу файла RFILE вместо указания значений OWNER:GROUP.

«-R, --recursive» - Обрабатывать каталоги и файлы рекурсивно

«-h, --help» - Отобразить краткую справку по команде и выйти.

«--version» - вывести информацию о версии и выйти

Следующие параметры изменяют способ обхода иерархии, если также указан параметр -R. Если указано более одного, вступает в силу только последний.

«-H» - если аргумент командной строки является символической ссылкой на каталог, пройти по ней

«-L» - пройти каждую символическую ссылку на встречающийся каталог

«-P» - не переходить по символическим ссылкам (по умолчанию)

chmod

```
chmod [OPTION]... MODE[,MODE]... FILE...  
chmod [OPTION]... OCTAL-MODE FILE...  
chmod [OPTION]... --reference=RFILE FILE...
```

Изменить биты прав доступа на файл.

chmod изменяет биты прав доступа каждого файла в соответствии с режимом, который может быть либо символическим представлением вносимых изменений, либо восьмеричным числом, представляющим битовую комбинацию для битов нового режима.

Формат символьного режима: [ugoа...][[-+=[perms...]]...], где perms — это либо ноль, либо несколько букв из набора «*gwxXst*», либо одна буква из набора. «*ugo*». Можно указать несколько символьных режимов, разделенных запятыми.

Комбинация букв «*ugoа*» определяет, доступ каких пользователей к файлу будет изменен: пользователь, владеющий им (*u*), другие пользователи из группы-владельца файла (*g*), прочие пользователи, не входящие в группу-владельца файла (*o*) или все пользователи (*a*). Если ни один из них не указан, эффект аналогичен заданному (*a*), но биты, установленные в *umask*, не затрагиваются.

Оператор *+* вызывает добавление выбранных битов режима файла к существующим битам режима файла каждого файла; *-* вызывает их удаление; *=* вызывает их добавление и удаление неупомянутых битов, за исключением того, что неупомянутые установленные биты идентификатора пользователя и группы каталога не затрагиваются.

Буквы «*gwxXst*» выбирают биты режима файла для затронутых пользователей: чтение (*r*), запись (*w*), выполнение (или поиск каталогов) (*x*), выполнение/поиск только в том случае, если файл является каталогом или уже имеет разрешение на выполнение для некоторых пользователь (*X*), установка идентификатора пользователя или группы при выполнении (*s*), флаг ограниченного удаления или бит закрепления (*t*). Вместо одной или нескольких из этих букв вы можете указать ровно одну из букв «*ugo*»: разрешения, предоставленные пользователю, владеющему файлом (*u*), разрешения, предоставленные другим пользователям, которые являются членами группы файла (*g*), и разрешения, предоставленные пользователям, которые не входят ни в одну из двух предыдущих категорий (*o*).

Числовой режим включает от одной до четырех восьмеричных цифр (0–7), полученных путем сложения битов со значениями 4, 2 и 1. Предполагается, что пропущенные цифры являются ведущими нулями. Первая цифра устанавливает распространение прав пользователя (4) и группы (2), на атрибуты ограниченного удаления или прикрепления (1). Вторая цифра выбирает разрешения для пользователя, которому принадлежит файл на чтение (4), запись (2) и выполнение (1); третий выбирает разрешения

для других пользователей в группе-владельце файла с теми же значениями; и четвертый для прочих пользователей, не входящих в группу-владельца файла, с теми же значениями.

`chmod` никогда не меняет разрешения символических ссылок; системный вызов `chmod` не может изменить их разрешения. Это не проблема, поскольку разрешения символических ссылок никогда не используются. Однако для каждой символической ссылки, указанной в командной строке, `chmod` изменяет права доступа к указанному файлу. Напротив, `chmod` игнорирует символические ссылки, встречающиеся во время рекурсивного обхода каталогов.

БИТЫ SETUID И SETGID

`chmod` очищает бит `set-group-ID` обычного файла, если идентификатор группы файла не соответствует эффективному идентификатору группы пользователя или одному из идентификаторов дополнительной группы пользователя, если только у пользователя нет соответствующих привилегий. Дополнительные ограничения могут привести к игнорированию битов `set-user-ID` и `set-group-ID` MODE или RFILE. Такое поведение зависит от политики и функциональности базового системного вызова `chmod`. В случае сомнений проверьте базовое поведение системы.

`chmod` сохраняет биты `set-user-ID` и `set-group-ID` каталога, если вы явно не укажете иное. Вы можете устанавливать или очищать биты в символьных режимах, таких как `u+s` и `g-s`, а также устанавливать (но не очищать) биты в числовом режиме.

ФЛАГ ОГРАНИЧЕННОГО УДАЛЕНИЯ ИЛИ ЛИПКИЙ БИТ

Флаг ограниченного удаления или липкий бит представляет собой один бит, интерпретация которого зависит от типа файла. Для каталогов он не позволяет непривилегированным пользователям удалять или переименовывать файлы в каталоге, если только они не являются владельцами файла или каталога; это называется флагом ограниченного удаления для каталога и обычно встречается в каталогах, доступных для записи всем, например `/tmp`. Для обычных файлов в некоторых старых системах этот бит сохраняет текстовое изображение программы на устройстве подкачки, поэтому при запуске оно загружается быстрее; это называется липким битом.

Параметры:

«-c, --changes» - подобно «-v», но выводит сообщения только при внесении изменений

«-f, --silent, --quiet» - подавляет большинство сообщений об ошибках

«-v, --verbose» - выводить диагностику для каждого обработанного файла

«--no-preserve-root» - не обрабатывать корневой каталог (по умолчанию)

«--preserve-root» - ошибка при рекурсивной обработке корневого каталога.

«--reference=RFILE» - использовать владельца и группу файла RFILE вместо

указания значений MODE.

«-R, --recursive» - Обрабатывать каталоги и файлы рекурсивно

«-h, --help» - Отобразить краткую справку по команде и выйти.

«--version» - вывести информацию о версии и выйти«xxx» -

umask

```
umask [-p] [-S] [mode]
```

Устанавливает маску создания пользовательского файла в режим «mode». Если режим начинается с цифры, он интерпретируется как восьмеричное число; в противном случае он интерпретируется как символическая маска режима, аналогичная той, которую принимает [chmod\(1\)](#). Если параметр mode опущен, печатается текущее значение маски. Опция -S приводит к печати маски в символической форме; вывод по умолчанию — восьмеричное число. Если указана опция -p и режим не указан, выходные данные имеют форму, которую можно повторно использовать в качестве входных данных.

setfacl

```
setfacl [-bkndRLPvh] [{-m|-x} acl_spec] [{-M|-X} acl_file] file ...  
setfacl --restore=file
```

Настраивает списки контроля доступа к файлам.

Эта утилита устанавливает списки контроля доступа (ACL) для файлов и каталогов. В командной строке за последовательностью команд следует последовательность файлов (за которой, в свою очередь, может следовать другая последовательность команд,...).

Опции -m и -x ожидают ACL в командной строке. Несколько записей ACL разделяются запятыми (,). Опции -M и -X считывают ACL из файла или из стандартного потока ввода. Формат записи ACL описан в разделе ЗАПИСИ ACL.

Опции --set и --set-file устанавливают ACL файла или каталога. Предыдущий ACL заменяется. Записи ACL для этой операции должны включать разрешения.

Опции -m (--modify) и -M (--modify-file) изменяют ACL файла или каталога. Записи ACL для этой операции должны включать разрешения.

Опции -x (--remove) и -X (--remove-file) удаляют записи ACL. Удаление несуществующей записи не является ошибкой. В качестве параметров принимаются только записи ACL без поля perms, если не определен POSIXLY_CORRECT.

При чтении файлов с использованием опций -M и -X setfacl принимает выходные данные, полученные getfacl. В каждой строке может быть не более одной записи ACL. После знака решетки («#») все, что находится до конца строки, рассматривается как комментарий.

Если setfacl используется в файловой системе, которая не поддерживает ACL, setfacl оперирует битами разрешения файлового режима. Если ACL не полностью соответствует

битам разрешений, `setfacl` изменяет биты разрешений файлового режима, чтобы они максимально точно отражали ACL, записывает сообщение об ошибке в стандартную ошибку и возвращается со статусом выхода больше 0.

РАЗРЕШЕНИЯ

Владельцу файла и процессам, поддерживающим `CAP_FOWNER`, предоставляется право изменять ACL файла. Это аналогично разрешениям, необходимым для доступа к файловому режиму. (В современных системах Linux `root` является единственным пользователем с возможностью `CAP_FOWNER`.)

ПАРАМЕТРЫ:

«`-b, --remove-all`» - Удаляет все расширенные записи ACL. Базовые записи ACL владельца, группы и других сохраняются.

«`-k, --remove-default`» - Удаляет ACL по умолчанию. Если ACL по умолчанию не существует, предупреждения не выдаются.

«`-n, --no-mask`» - Не пересчитывать эффективную маску прав. По умолчанию `setfacl` пересчитывает запись маски ACL, если только запись маски не была указана явно. Запись маски настроена на объединение всех разрешений группы-владельца, а также всех именованных записей пользователей и групп. (Это именно те записи, на которые влияет запись маски).

«`--mask`» - Пересчитывает действующую маску прав, даже если запись маски ACL была явно задана. (См. параметр `-n`.)

«`-d, --default`» - Все операции применяются к ACL по умолчанию. Обычные записи ACL во входном наборе повышаются до записей ACL по умолчанию. Записи ACL по умолчанию во входном наборе отбрасываются. (В этом случае выдается предупреждение).

«`--restore=file`» - Восстанавливает резервную копию разрешений, созданную с помощью `'getfacl -R'` или аналогичного. Все разрешения всего поддерева каталога восстанавливаются с использованием этого механизма. Если входные данные содержат комментарии владельца или группы, `setfacl` пытается восстановить владельца и группу-владельца. Если входные данные содержат комментарии к флагам (которые определяют биты `setuid`, `setgid` и `Sticky`), `setfacl` соответственно устанавливает эти три бита; в противном случае он очищает их. Эту опцию нельзя смешивать с другими опциями, кроме `'--test'`.

«`--test`» - Режим тестирования. Вместо изменения списков ACL каких-либо файлов отображаются полученные списки ACL.

«`-R, --recursive`» - Рекурсивно применять операции ко всем файлам и каталогам. Эту опцию нельзя смешивать с `--restore`.

«`-L, --logical`» - Логическая прогулка, переход по символическим ссылкам на

каталоги. Поведение по умолчанию — следовать символическим ссылкам, заданным в непосредственно команде и пропускать символические ссылки, встречающиеся в подкаталогах. Эффективен только в сочетании с -R. Эту опцию нельзя смешивать с --restore.

«-P, --physical» - Физическая прогулка, не переходит по символическим ссылкам на каталоги. Также пропускает символической ссылки, перечисленные непосредственно в команде. Эффективен только в сочетании с -R. Эту опцию нельзя смешивать с --restore.

«--version» - вывести информацию о версии и выйти

«-h, --help» - Отобразить краткую справку по команде и выйти.

«--» - Конец параметров командной строки. Все остальные параметры интерпретируются как имена файлов, даже если они начинаются с тире.

«-» - Если параметр имени файла представляет собой одно тире, setfacl считывает список файлов из стандартного ввода

ЗАПИСИ ACL

Утилита setfacl распознает следующие форматы записей ACL (пробелы вставлены для ясности):

```
[d[efault]:] [u[ser]:]uid [:perms]
```

Разрешения именованного пользователя. Разрешения владельца файла, если uid пуст.

```
[d[efault]:] g[roup]:gid [:perms]
```

Разрешения именованной группы. Разрешения группы-владельца, если gid пуст.

```
[d[efault]:] m[ask][:] [:perms]
```

Эффективная маска прав

```
[d[efault]:] o[ther][:] [:perms]
```

Разрешения для прочих пользователей.

Пробелы между символами-разделителями и символами, не являющимися разделителями, игнорируются.

Правильные записи ACL, включая разрешения, используются в операциях изменения и установки. (опции -m, -M, --set и --set-file). Записи без поля разрешений используются для удаления записей (параметры -x и -X).

Для uid и gid вы можете указать имя или номер. Символьные литералы могут быть указаны с помощью обратной косой черты, за которой следуют трехзначные восьмеричные цифры, соответствующие ASCII-коду символа (например, \101 для 'A'). Если имя содержит буквальную обратную косую черту, за которой следуют 3 цифры, обратную косую черту необходимо экранировать (т. е. \\).

Поле perms представляет собой комбинацию символов, обозначающих разрешения на чтение (r), запись (w), выполнение (x). Символы тире в поле разрешений (-) игнорируются.

Символ X обозначает разрешение на выполнение, если файл является каталогом или уже имеет разрешение на выполнение для какого-либо пользователя. Альтернативно, поле perms может определять разрешения в числовом виде в виде побитовой комбинации чтения (4), записи (2) и выполнения (1). Поля с нулевым разрешением или поля разрешений, состоящие только из тире, означают отсутствие разрешений.

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/>