Отчет
Лабораторная работа по Linux

Выполнила: Рымкулова Диана СКБ181

1. Создание пользователей. Задание паролей. Сброс пароля пользователя.

a. Создать две учетные записи пользователей: user1, user2.

```
rymkulova-diana@ubuntu:~/Desktop$ sudo useradd user1
rymkulova-diana@ubuntu:~/Desktop$ sudo useradd user2
rymkulova-diana@ubuntu:~/Desktop$
```

Созданы две учетные записи с помощью команды useradd

b. Задать пользователям одинаковые пароли.

```
rymkulova-diana@ubuntu:~/Desktop$ sudo passwd user1
New password:
Retype new password:
passwd: password updated successfully
rymkulova-diana@ubuntu:~/Desktop$ sudo passwd user2
New password:
Retype new password:
passwd: password updated successfully
rymkulova-diana@ubuntu:~/Desktop$
```

Заданы одинаковые пароли pass1234 с помощью команды passwd

c. Проанализировать файл /etc/shadow и /etc/passwd. Сделать соответствующие выводы.

```
rymkulova-diana@ubuntu:~/Desktop$ cat/etc/passwd | grep user
bash: cat/etc/passwd: No such file or directory
rymkulova-diana@ubuntu:~/Desktop$ cat /etc/passwd | grep user
cups-pk-helper:x:115:122:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
sssd:x:118:125:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin
hplip:x:126:7:HPLIP system user,,,:/run/hplip:/bin/false
user1:x:1001:1001::/home/user1:/bin/sh
user2:x:1002:1002::/home/user2:/bin/sh
rymkulova-diana@ubuntu:~/Desktop$ sudo cat /etc/shadow | grep user
user1:$y$j9T$sUugPRjfZNIl/JaD.LxR2.$cLfyceEmbhDR8IP4aJYKBF6pF1kPf8tdhidboqg5ds.:19315:0:99999:7:::
user2:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
rymkulova-diana@ubuntu:~/Desktop$
```

Файл /etc/shadow доступен для чтения только пользователю root и предназначен для хранения зашифрованных паролей. В нем также содержится учетная информация, которая отсутствует в файле /etc/passwd, например, срок действия учетной записи, количество дней по истечении срока действия пароля и др.
Помимо этого, стоит отметить, что даже в нашем случае, когда пароли пользователей совпадают, хеши паролей отличаются. Это связано с тем, что к самому паролю добавляется различная соль, а затем происходит процесс хеширования.

d. Из файла /etc/shadow удалить свертку пароля пользователя user1.

```
rymkulova-diana@ubuntu:~/Desktop$ sudo cat /etc/shadow | grep user
user1:$y$j9T$sUugPRjfZNIl/JaD.LxR2.$cLfyceEmbhDR8IP4aJYKBF6pF1kPf8tdhidboqg5ds.:19315:0:99999:7:::
user2:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
rymkulova-diana@ubuntu:~/Desktop$ sudo nano /etc/shadow
rymkulova-diana@ubuntu:~/Desktop$ sudo cat /etc/shadow | grep user
user1::19315:0:99999:7:::
user2:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
rymkulova-diana@ubuntu:~/Desktop$
```

Для того, чтобы удалить свертку пароля воспользовалась текстовым редактором nano.

e. Проверить, каким образом user1 войдет в систему. Сделать выводы.

```
rymkulova-diana@ubuntu:~/Desktop$ su user1
$ whoami
user1
$ exit
rymkulova-diana@ubuntu:~/Desktop$ sudo cat /etc/shadow | grep user
user1::19315:0:99999:7:::
user2:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
rymkulova-diana@ubuntu:~/Desktop$
```

Войти под user1 без какого-либо пароля получилось, соответственно в Ubuntu 22.04 такой вход можно реализовать.

f. В файле /etc/shadow заменить свертку пароля для пользователя user1 сверткой пароля user2.

```
rymkulova-diana@ubuntu:~/Desktop$ sudo cat /etc/shadow | grep user
user1::19315:0:99999:7:::
user2:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
rymkulova-diana@ubuntu:~/Desktop$ sudo nano /etc/shadow
rymkulova-diana@ubuntu:~/Desktop$ sudo cat /etc/shadow | grep user
user1:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
user2:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
rymkulova-diana@ubuntu:~/Desktop$
```

Также, как и в пункте (d) для замены свертки воспользовалась текстовым редактором nano.

g. Проверить, каким образом user1 войдет в систему. Сделать выводы.

```
rymkulova-diana@ubuntu:~/Desktop$ su user1
Password:
$ whoami
user1
$
```

Теперь для входа под пользователем user1 снова необходим ввод пароля, следовательно пароль пользователя можно изменять вручную в файле /etc/shadow.

2. Создание пользователей вручную.

a. Вручную (без использования команды useradd или adduser) добавить пользователя user3.

```
rymkulova-diana@ubuntu:~/Desktop$ sudo nano /etc/passwd
rymkulova-diana@ubuntu:~/Desktop$ sudo cat /etc/passwd | grep user
cups-pk-helper:x:115:122:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
sssd:x:118:125:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin
hplip:x:126:7:HPLIP system user,,,:/run/hplip:/bin/false
user1:x:1001:1001::/home/user1:/bin/sh
user2:x:1002:1002::/home/user2:/bin/sh
user3:x:1003:1003::/home/user3:/bin/sh
rymkulova-diana@ubuntu:~/Desktop$
```

User3 создан с помощью текстового редактора nano.

b. Пароль пользователя задать вручную (без использования команды passwd).

Для начала был сформирован хэш пароля с помощью алгоритма хеширования SHA-512

```
rymkulova-diana@ubuntu:~/Desktop$ openssl passwd -6 -salt salt user
$6$salt$eJAWjOhOA5txn6hiwB8Mbhe/D9/7I09LnEuJ.uo.KsuLd1c0cuDmF4LMble.0hCLnjeJU9C3cYcvmtctzlCir/
```

Затем с помощью nano хэш пароля добавлен в файл /etc/shadow.

```
rymkulova-diana@ubuntu:~/Desktop$ sudo cat /etc/shadow | grep user
user1:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
user2:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
user3:
rymkulova-diana@ubuntu:~/Desktop$ sudo nano /etc/shadow
rymkulova-diana@ubuntu:~/Desktop$ sudo cat /etc/shadow | grep user
user1:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
user2:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
user3:$6$salt$eJAWjOhOA5txn6hiwB8Mbhe/D9/7I09LnEuJ.uo.KsuLd1c0cuDmF4LMble.0hCLnjeJU9C3cYcvmtctzlCir/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:193
15:0:99999:7:::
rymkulova-diana@ubuntu:~/Desktop$
```

c. Задать ограничения на пароль вручную, время действия пароля 3 дня (без использования команды passwd).

Также в /etc/shadow заменяем время действия пароля, по умолчанию это 99999, заменяем на 3

```
rymkulova-diana@ubuntu:~/Desktop$ sudo cat /etc/shadow | grep user
user1:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
user2:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
user3:$6$salt$eJAWjOhOA5txn6hiwB8Mbhe/D9/7I09LnEuJ.uo.KsuLd1c0cuDmF4LMble.0hCLnjeJU9C3cYcvmtctzlCir/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:3:7:::
rymkulova-diana@ubuntu:~/Desktop$
```

d. Задать ограничения на пароль вручную, предупреждать о смене пароля за 5 дней (без использования команды passwd), убедиться в наличии предупреждений.

Также в /etc/shadow заменяем время действия пароля, по умолчанию это 7, заменяем на 5

```
rymkulova-diana@ubuntu:~/Desktop$ sudo nano /etc/shadow
rymkulova-diana@ubuntu:~/Desktop$ sudo cat /etc/shadow | grep user
user1:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
user2:$y$j9T$WOwlmlSR0iPJkZDUGj9QZ0$nnPvYx3mmJ6N3X/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:99999:7:::
user3:$6$salt$eJAWjOhOA5txn6hiwB8Mbhe/D9/7I09LnEuJ.uo.KsuLd1c0cuDmF4LMble.0hCLnjeJU9C3cYcvmtctzlCir/HKnK1PPYMbvrBx4Ysozoq.xJX6v5:19315:0:3:5:::
rymkulova-diana@ubuntu:~/Desktop$
```

```
rymkulova-diana@ubuntu:~/Desktop$ su user3
Password:
Warning: your password will expire in 3 days.
```

Так как количество дней действия пароля меньше, чем дней для предупреждения, то появляется уведомление о 3 днях (пароль создан сегодня) и будет появляться всегда при подобных настройках.

3. Добавление пользователей в привилегированную группу (sudoers).
a. Добавить пользователю user3 возможность выполнять команды от имени пользователя user1 с запросом пароля.

Для того, чтобы добавить user3 в sudoers воспользовалась текстовым редактором через команду visudo

```
rymkulova-diana@ubuntu:~/Desktop$ sudo visudo
rymkulova-diana@ubuntu:~/Desktop$ sudo cat /etc/sudoers | grep user3
user3 ALL=(user1) ALL
rymkulova-diana@ubuntu:~/Desktop$
```

b. Убедиться в возможности выполнения команд от имени пользователя user1

```
rymkulova-diana@ubuntu:~/Desktop$ su user3
Password:
Warning: your password will expire in 3 days.
$ sudo -u user1 whoami
[sudo] password for user3:
user1
$
```

c. Добавить пользователю user3 возможность выполнять команды от имени пользователя user2 без запроса пароля.

Аналогично пункту (a) редактируем возможности

```
rymkulova-diana@ubuntu:~/Desktop$ sudo visudo
rymkulova-diana@ubuntu:~/Desktop$ sudo cat /etc/sudoers | grep user3
user3 ALL=(user1) ALL
user3 ALL=(user2)NOPASSWD: ALL
```

Пароль не запрашивается

```
rymkulova-diana@ubuntu:~/Desktop$ su user3
Password:
Warning: your password will expire in 3 days.
$ sudo -u user2 whoami
user2
```

4. Разграничение прав пользователей.
a. Создать двух пользователей user1 и user2.

Пользователи user1 и user2 созданы в рамках задачи 1.

b. В директории /tmp создать файл file.

```
rymkulova-diana@ubuntu:/$ sudo su
root@ubuntu:/# touch /tmp/file
```

c. Настроить его ACL таким образом, чтобы user1 имел полный доступ к файлу, а user2 мог только читать из него.

```
rymkulova-diana@ubuntu:/$ sudo su
root@ubuntu:/# touch /tmp/file
root@ubuntu:/# setfacl -m u:user1:rwx /tmp/file
root@ubuntu:/# setfacl -m u:user2:r /tmp/file
root@ubuntu:/# getfacl /tmp/file
getfacl: Removing leading '/' from absolute path names
# file: tmp/file
# owner: root
# group: root
user::rw-
user:user1:rwx
user:user2:r--
group::r--
mask::rwx
other::r--

root@ubuntu:/#
```

d. Убедиться, что права настроены правильно, для этого записать от имени user1 данные файл, а затем считать их от имени user2. Затем попробовать записать от имени user2 и убедиться, что это сделать невозможно.

В файл записано hello под user1

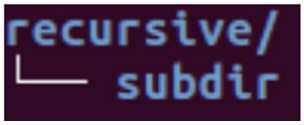
```
rymkulova-diana@ubuntu:~/Desktop$ su user1
Password:
$ echo "hello" > /tmp/file
$ cat /tmp/file
hello
$
```

user2 не имеет возможности сделать запись в файл, но может прочитать уже имеющуюся

```
rymkulova-diana@ubuntu:~/Desktop$ su user2
Password:
$ echo "hello x2" >> /tmp/file
sh: 1: cannot create /tmp/file: Permission denied
$ cat /tmp/file
hello
$
```

5. Рекурсивная настройка прав директорий.

a. В директории /tmp создать следующую структуру файлов:




```
rymkulova-diana@ubuntu:/tmp$ sudo su
root@ubuntu:/tmp# mkdir -p /tmp/recursive/subdir
root@ubuntu:/tmp# tree recursive
recursive
└── subdir

1 directory, 0 files
root@ubuntu:/tmp#
```

b. Рекурсивно установить ACL права на всю указанную выше структуру так, чтобы user1 мог писать в каждую поддиректорию.

```
root@ubuntu:/tmp# setfacl -R -m u:user1:w /tmp/recursive
root@ubuntu:/tmp# getfacl /tmp/recursive
getfacl: Removing leading '/' from absolute path names
# file: tmp/recursive
# owner: root
# group: root
user::rwx
user:user1:-w-
group::r-x
mask::rwx
other::r-x

root@ubuntu:/tmp#
```

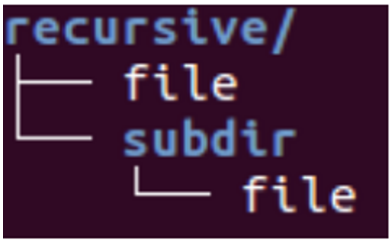
Установили права и проверили в /tmp/recursive

Также проверили в /tmp/recursive/subdir

```
root@ubuntu:/tmp# getfacl /tmp/recursive/subdir
getfacl: Removing leading '/' from absolute path names
# file: tmp/recursive/subdir
# owner: root
# group: root
user::rwx
user:user1:-w-
group::r-x
mask::rwx
other::r-x

root@ubuntu:/tmp#
```

c. Убедится в правильности установки прав, создав следующую структуру от имени user1:




```
user1@ubuntu:/tmp$ touch recursive/file
touch: cannot touch 'recursive/file': Permission denied
user1@ubuntu:/tmp$ touch recursive/subdir/file
touch: cannot touch 'recursive/subdir/file': Permission denied
user1@ubuntu:/tmp$
```

Мы не можем создать такую структуру от имени user1 с правами установленными в рамках пункта (b), так как необходимы права на выполнение, добавив их, получим:

```
root@ubuntu:/tmp# setfacl -R -m u:user1:wx /tmp/recursive
root@ubuntu:/tmp#
```

```
user1@ubuntu:/tmp$ touch recursive/subdir/file
user1@ubuntu:/tmp$ touch recursive/file
user1@ubuntu:/tmp$
```

```
root@ubuntu:/tmp# tree recursive
recursive
├── file
└── subdir
    └── file

1 directory, 2 files
root@ubuntu:/tmp#
```

6. ACL по умолчанию.

a. В директории /tmp создать поддиректорию test.

```
rymkulova-diana@ubuntu:/tmp$ mkdir test
rymkulova-diana@ubuntu:/tmp$
```

b. Установить на эту директорию ACL по умолчанию таким образом, чтобы user1 мог только читать файлы, размещенные в нём, а user2 мог только записывать в файлы в нём.

```
rymkulova-diana@ubuntu:/tmp$ mkdir test
rymkulova-diana@ubuntu:/tmp$ sudo su
[sudo] password for rymkulova-diana:
root@ubuntu:/tmp# setfacl -m d:u:user1:r test
root@ubuntu:/tmp# setfacl -m d:u:user2:w test
root@ubuntu:/tmp# getfacl test
# file: test
# owner: rymkulova-diana
# group: rymkulova-diana
user::rwx
group::rwx
other::r-x
default:user::rwx
default:user:user1:r--
default:user:user2:-w-
default:group::rwx
default:mask::rwx
default:other::r-x

root@ubuntu:/tmp#
```

c. Убедиться, что права настроены правильно, для этого создать файл file в этой директории и попробовать записать в него данные сначала от имени user1, убедиться, что это невозможно, а затем от имени user2. Аналогично, попробовать считать данные по очереди за каждого из созданных пользователей.

```
root@ubuntu:/tmp# cd test
root@ubuntu:/tmp/test# touch file
```

```
rymkulova-diana@ubuntu:/tmp$ cd test
rymkulova-diana@ubuntu:/tmp/test$ su user1
Password:
$ bash
user1@ubuntu:/tmp/test$ echo hello > file
bash: file: Permission denied
```

```
rymkulova-diana@ubuntu:/tmp/test$ su user2
Password:
$ bash
user2@ubuntu:/tmp/test$ echo hello > file
user2@ubuntu:/tmp/test$ cat file
cat: file: Permission denied
```

```
rymkulova-diana@ubuntu:/tmp/test$ su user1
Password:
$ cat file
hello
$
```

d. Создать ещё один файл file2 в tmp. Установить его права в ACL так, чтобы user2 мог из него читать. Убедиться, что user2 имеет возможность читать из file2. Для этого от имени user2 записать в него данные, а затем вывести его содержимое на экран.

```
rymkulova-diana@ubuntu:/tmp$ sudo su
root@ubuntu:/tmp# touch file2
root@ubuntu:/tmp# setfacl -m u:user2:r file2
root@ubuntu:/tmp# su user2
$ echo > file2
sh: 1: cannot create file2: Permission denied
$ cat file2
$
```

Нет возможности записать что-либо в этот файл, так как нет соответствующих прав, читать возможность есть, но файл пустой.

На протяжении всего этого пункта мы работали в директории test, может быть файл создавать необходимо было в ней? Если это так, то получается похожий результат:

```
root@ubuntu:/tmp# cd test
root@ubuntu:/tmp/test# touch file2
root@ubuntu:/tmp/test# setfacl -m u:user2:r file2
root@ubuntu:/tmp/test# su user2
$ echo > file2
sh: 1: cannot create file2: Permission denied
$ cat file2
$
```

Из чего можно сделать, что права перезаписываются.

7. Эффективная маска.

a. Создать в директории /tmp файл mask и записать в него произвольный текст.

```
rymkulova-diana@ubuntu:/tmp$ echo hello > mask
rymkulova-diana@ubuntu:/tmp$
```

b. Модифицировать ACL: дать пользователю user1 право на чтение и запись в mask.

```
rymkulova-diana@ubuntu:/tmp$ setfacl -m u:user1:rw mask
rymkulova-diana@ubuntu:/tmp$ getfacl mask
# file: mask
# owner: rymkulova-diana
# group: rymkulova-diana
user::rw-
user:user1:rw-
group::rw-
mask::rw-
other::r--

rymkulova-diana@ubuntu:/tmp$
```

c. Установить в ACL этого файла эффективную маску так, чтобы никто не мог записывать в файл.

```
rymkulova-diana@ubuntu:/tmp$ setfacl -m m:r mask
```

d. Убедиться в том, что user1 не может ничего записать в mask, но может из него считать.

```
rymkulova-diana@ubuntu:/tmp$ setfacl -m m:r mask
rymkulova-diana@ubuntu:/tmp$ su user1
Password:
$ echo hello > mask
sh: 1: cannot create mask: Permission denied
$
```

```
rymkulova-diana@ubuntu:/tmp$ su user1
Password:
$ cat mask
hello
$
```

8. Копирование ACL.

a. Создать в директории /tmp файлы source и dest и записать в них текстовую информацию. Установить этим файлам стандартные UNIX-права 660, чтобы user1 и user2 не имели доступа к файлам.

```
rymkulova-diana@ubuntu:/tmp$ echo hello > source
rymkulova-diana@ubuntu:/tmp$ echo hello > dest
rymkulova-diana@ubuntu:/tmp$ chmod 660 source
rymkulova-diana@ubuntu:/tmp$ chmod 660 dest
```

b. Настроить ACL правила source так, чтобы user1 мог читать из него, а правила dest так, чтобы из него мог читать user2.

```
rymkulova-diana@ubuntu:/tmp$ setfacl -m u:user1:r source
rymkulova-diana@ubuntu:/tmp$ setfacl -m u:user2:r dest
rymkulova-diana@ubuntu:/tmp$
```

c. Убедиться, что каждый из пользователей может читать из соответствующего файла.

```
rymkulova-diana@ubuntu:/tmp$ su user1
Password:
$ cat source
hello
$ su user2
Password:
$ cat dest
hello
$
```

d. Скопировать ACL из файла source в файл dest.

```
rymkulova-diana@ubuntu:/tmp$ getfacl source | setfacl --set-file=- dest
rymkulova-diana@ubuntu:/tmp$
```

e. Убедиться, что из файла dest может читать только пользователь user1.

```
rymkulova-diana@ubuntu:/tmp$ su user1
Password:
$ cat dest
hello
$ su user2
Password:
$ cat dest
cat: dest: Permission denied
$
```

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Основные команды для работы с пользователями.

sudo, su, adduser,useradd, passwd, deluser,userdel, groups, addgroup, groupadd

2. Почему нужны два разных файла /etc/passwd и /etc/shadow, почему нельзя использовать один из них?

etc/passwd содержит основную информацию о каждой учетной записи пользователя в системе. Раньше etc/passwd хранил в себе зашифрованный пароль. В настоящее время содержит в себе букву x, чтобы обозначить, что пароль был назначен, но был сохранен в другом файле теневого файла.

etc/shadow хранит в себе фактические пароли пользователей в зашифрованном формате. На самом деле, есть хэш пароля с дополнительными свойствами, относящиеся к паролям пользователя, такие как даты истечения срока действия пароля.

Файл etc/passwd доступен для чтения по словам, а это означает, что любой пользователь может его прочитать, но файл etc/shadow доступен для чтения только учетной записи root.

Существование двух файлов позволяет лучше обеспечивать безопасность в системе, так как зашифрованный пароль не находится в файле (с хранимой в открытом виде солью), к которому может получить кто-то доступ (если он не root).

3. Зачем нужны SUID и SGID и Stikybit?

Когда бит SUID установлен для исполняемого файла, это означает, что файл будет выполняться с теми же разрешениями, что и владелец исполняемого файла.

С установленным битом SGID любой пользователь, выполняющий файл, будет иметь те же права, что и владелец группы файла. Это преимущество в обращении с каталогом. Когда разрешение SGID применяется к каталогу, все подкаталоги и файлы, созданные в этом каталоге, получат то же владение группой, что и основной каталог (а не владение группой пользователя, создавшего файлы и каталоги).

Stikybit работает над каталогом. Если для каталога установлен Stikybit, все файлы в каталоге могут быть удалены или переименованы только владельцами файлов или пользователем root.

4. Зачем в Linux были введены списки контроля доступа?

В Linux есть специфика разрешений, из-за ограничений настройки прав, владения на файлы, папки в Linux (например, мы не можем настроить разные права доступа для разных пользователей к одному и тому же файлу). Чтобы преодолеть эту проблему, введены ACL.

5. Какие базовые утилиты используются для управления ACL?

setfacl — утилита, которая предназначена для установки, модификации и удаления ACL
getfacl — утилита, которая предназначена для получения информации об установленных ACL.

6. Зачем нужны ACL по умолчанию?

Для директории можно указать ACL права по умолчанию, которые будут автоматически добавляться для файлов и директорий, создаваемых в ней.

7. Как понять, что для файла установлен ACL?

Вызвать команду ls -l, если ACL установлен, то в конце файла будет символ +.

8. Чем лучше воспользоваться, когда необходимо разрешить выполнение конкретного исполняемого файла конкретному пользователю, ACL или прописать правило в sudoers?

Я считаю, что лучше воспользоваться ACL, позволяют он нам предоставлять разные права доступа разным пользователям, а также предоставлять доступ без необходимости возиться с фактическими базовыми разрешениями файла или папки.
В случае же sudoers, мы ,конечно, можем там прописать правило, но выглядит это не очень целесообразно, так как риски (например, редактирование файла sudoers с ошибками или неправильным синтаксисом может привести к блокировке всех пользователей в вашем дистрибутиве, этого можно постараться избежать используя visudo, но и это не панацея) что-то сделать не так будут скорее всего выше, чем полезное действие.