Отчёт по лабораторной работе №3

Настройка прав доступа

Ришард Когенгар

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Получение навыков настройки базовых и специальных прав доступа для групп пользователей в операционной системе типа Linux.

2 Ход выполнения

2.1 Управление базовыми разрешениями доступа

1. В терминале выполнен вход под учётной записью суперпользователя **root** с помощью команды `su -`, что необходимо для выполнения операций администрирования файловой системы.

2. В корневом каталоге файловой системы созданы два каталога: **/data/main** и **/data/third**.
   После создания выполнена проверка владельца и группы каталогов командой `ls -Al /data`.
   На данном этапе владельцем и группой обоих каталогов являлся пользователь **root**.

```
rishard@rishardkogengar:~$ su
Password:
root@rishardkogengar:/home/rishard#
root@rishardkogengar:/home/rishard# mkdir -p /data/main /data/third
root@rishardkogengar:/home/rishard# ls -Al /data
total 0
drwxr-xr-x. 2 root root 6 Jan 18 13:04 main
drwxr-xr-x. 2 root root 6 Jan 18 13:04 third
root@rishardkogengar:/home/rishard# chgrp main /data/main
root@rishardkogengar:/home/rishard# chgrp third /data/third/
root@rishardkogengar:/home/rishard# ls -Al /data
total 0
drwxr-xr-x. 2 root main  6 Jan 18 13:04 main
drwxr-xr-x. 2 root third 6 Jan 18 13:04 third
root@rishardkogengar:/home/rishard# chmod 770 /data/main/
root@rishardkogengar:/home/rishard# chmod 770 /data/third/
root@rishardkogengar:/home/rishard# ls -Al /data
total 0
drwxrwx---. 2 root main  6 Jan 18 13:04 main
drwxrwx---. 2 root third 6 Jan 18 13:04 third
root@rishardkogengar:/home/rishard#
```

Рис. 2.1: Создание каталогов и проверка владельцев

3. Для разграничения прав доступа изменены группы-владельцы каталогов:

   - для каталога **/data/main** назначена группа **main**;
   - для каталога **/data/third** назначена группа **third**.

   Повторная проверка с помощью `ls -Al /data` подтвердила, что группы-владельцы изменены корректно.

4. Установлены разрешения, позволяющие владельцу и группе записывать в каталоги и запрещающие доступ всем остальным пользователям: режим `770` для **/data/main** и **/data/third**.
   Это соответствует правам `drwxrwx---`.
   Проверка подтвердила корректность установки прав.

5. В отдельном терминале выполнен вход под пользователем **bob** (пользова-

тель является членом группы **main**).

6. Под пользователем **bob** выполнен переход в каталог **/data/main** и создан файл **emptyfile**.
   Операция завершилась успешно, так как группа **main** имеет права записи в каталог **/data/main**.

```
root@rishardkogengar:/home/rishard#
root@rishardkogengar:/home/rishard# su bob
bob@rishardkogengar:/home/rishard$ cd /data/main/
bob@rishardkogengar:/data/main$ touch emptyfile
bob@rishardkogengar:/data/main$ ls -Al
total 0
-rw-r--r--. 1 bob bob 0 Jan 18 13:06 emptyfile
bob@rishardkogengar:/data/main$ cd /data/third/
bash: cd: /data/third/: Permission denied
bob@rishardkogengar:/data/main$
```

Рис. 2.2: Создание файла пользователем bob в /data/main

7. Под пользователем **bob** выполнена попытка перейти в каталог **/data/third**. Получен отказ в доступе (**Permission denied**), поскольку пользователь **bob** не является членом группы **third**, а права для остальных пользователей отсутствуют.

2.2 Управление специальными разрешениями (setgid и sticky bit)

8. В новом терминале выполнен вход под пользователем **alice** (также член группы **main**).
   В каталоге **/data/main** созданы файлы **alice1** и **alice2**.

9. В другом терминале выполнен вход под пользователем **bob**, после чего в каталоге **/data/main** выполнен просмотр списка файлов.

Далее пользователь **bob** удалил файлы, созданные пользователем **alice**. Удаление произошло успешно, так как на данном этапе отсутствовал sticky bit, а запись в каталог группе была разрешена.

```
bob@rishardkogengar:/data/main$
bob@rishardkogengar:/data/main$ su alice
Password:
alice@rishardkogengar:/data/main$ cd /data/main/
alice@rishardkogengar:/data/main$ touch alice1
alice@rishardkogengar:/data/main$ touch alice2
alice@rishardkogengar:/data/main$ su bob
Password:
bob@rishardkogengar:/data/main$ cd /data/main/
bob@rishardkogengar:/data/main$ ls -l
total 0
-rw-r--r--. 1 alice alice 0 Jan 18 13:08 alice1
-rw-r--r--. 1 alice alice 0 Jan 18 13:08 alice2
-rw-r--r--. 1 bob   bob   0 Jan 18 13:06 emptyfile
bob@rishardkogengar:/data/main$ rm -f alice*
bob@rishardkogengar:/data/main$ ls -l
total 0
-rw-r--r--. 1 bob bob 0 Jan 18 13:06 emptyfile
bob@rishardkogengar:/data/main$
```

Рис. 2.3: Удаление файлов alice пользователем bob

10. Пользователем **bob** созданы файлы **bob1** и **bob2** в каталоге **/data/main**.

11. Под пользователем **root** для каталога **/data/main** установлены:

    - бит идентификатора группы (**setgid**), обеспечивающий наследование группы-владельца каталога всеми новыми файлами;
    - **sticky bit**, запрещающий удаление/переименование файлов пользователями, не являющимися владельцами этих файлов.

12. Под пользователем **alice** в каталоге **/data/main** созданы файлы **alice3** и **alice4**.

    Проверка показала, что группа новых файлов стала **main**, то есть setgid сработал корректно.

13. Пользователь **alice** попытался удалить файлы **bob1** и **bob2**, принадлежащие пользователю **bob**.

    Удаление было запрещено, что подтверждает работу sticky bit: даже при наличии прав записи в каталог удалять можно только свои файлы (или при наличии административных прав).

```
bob@rishardkogengar:/data/main$ touch bob1
bob@rishardkogengar:/data/main$ touch bob2
bob@rishardkogengar:/data/main$ su
Password:
root@rishardkogengar:/data/main# chmod g+s,o+t /data/main/
root@rishardkogengar:/data/main# su alice
alice@rishardkogengar:/data/main$ touch alice3
alice@rishardkogengar:/data/main$ touch alice4
alice@rishardkogengar:/data/main$ ls -l
total 0
-rw-r--r--. 1 alice main 0 Jan 18 13:10 alice3
-rw-r--r--. 1 alice main 0 Jan 18 13:10 alice4
-rw-r--r--. 1 bob   bob  0 Jan 18 13:10 bob1
-rw-r--r--. 1 bob   bob  0 Jan 18 13:10 bob2
-rw-r--r--. 1 bob   bob  0 Jan 18 13:06 emptyfile
alice@rishardkogengar:/data/main$ rm -rf bob*
rm: cannot remove 'bob1': Operation not permitted
rm: cannot remove 'bob2': Operation not permitted
alice@rishardkogengar:/data/main$
```

Рис. 2.4: Действие sticky bit: запрет удаления чужих файлов

2.3 Управление расширенными разрешениями (ACL)

14. Под пользователем **root** установлены расширенные разрешения ACL:

    - группе **third** предоставлены права чтения и выполнения в каталоге **/data/main**;
    - группе **main** предоставлены права чтения и выполнения в каталоге **/data/third**.

15. Командой `getfacl` подтверждено, что ACL-записи применены корректно для обоих каталогов.

```
alice@rishardkogengar:/data/main$ su
Password:
root@rishardkogengar:/data/main# setfacl -m g:third:rx /data/main
root@rishardkogengar:/data/main# setfacl -m g:main:rx /data/third
root@rishardkogengar:/data/main# getfacl /data/main
getfacl: Removing leading '/' from absolute path names
# file: data/main
# owner: root
# group: main
# flags: -st
user::rwx
group::rwx
group:third:r-x
mask::rwx
other::---

root@rishardkogengar:/data/main# getfacl /data/third
getfacl: Removing leading '/' from absolute path names
# file: data/third
# owner: root
# group: third
user::rwx
group::rwx
group:main:r-x
mask::rwx
other::---

root@rishardkogengar:/data/main#
```

Рис. 2.5: Проверка ACL каталогов

16. В каталоге **/data/main** создан файл **newfile1**, после чего выполнена проверка его ACL.

    Установленные для каталога ACL-права не распространились на файл автоматически, так как ACL по умолчанию (default ACL) на каталоге ещё не были настроены.

    Поэтому права файла определились стандартным механизмом: владельцем, основной группой и базовой маской прав.

Аналогичное поведение наблюдалось при создании файла **newfile1** в каталоге **/data/third**.

```
root@rishardkogengar:/data/main# touch /data/main/newfile1
root@rishardkogengar:/data/main# getfacl /data/main/newfile1
getfacl: Removing leading '/' from absolute path names
# file: data/main/newfile1
# owner: root
# group: main
user::rw-
group::r--
other::r--

root@rishardkogengar:/data/main# touch /data/third/newfile1
root@rishardkogengar:/data/main# getfacl /data/third/newfile1
getfacl: Removing leading '/' from absolute path names
# file: data/third/newfile1
# owner: root
# group: root
user::rw-
group::r--
other::r--

root@rishardkogengar:/data/main#
```

Рис. 2.6: Наследование ACL новыми файлами

17. Для обеспечения наследования ACL установлены ACL по умолчанию:

    - для каталога **/data/main** добавлены default-права для группы **third**;
    - для каталога **/data/third** добавлены default-права для группы **main**.

18. После настройки default ACL созданы файлы **newfile2** в каталогах **/data/main** и **/data/third**.

    Проверка показала, что новые файлы получили наследуемые ACL-записи (через default ACL каталога), что подтверждает корректность настройки наследования.

```
root@rishardkogengar:/data/main# setfacl -m d:g:third:rwx /data/main/
root@rishardkogengar:/data/main# setfacl -m d:g:main:rwx /data/third/
root@rishardkogengar:/data/main# touch /data/main/newfile2
root@rishardkogengar:/data/main# getfacl /data/main/newfile2
getfacl: Removing leading '/' from absolute path names
# file: data/main/newfile2
# owner: root
# group: main
user::rw-
group::rwx                      #effective:rw-
group:third:rwx                 #effective:rw-
mask::rw-
other::---

root@rishardkogengar:/data/main# touch /data/third/newfile2
root@rishardkogengar:/data/main# getfacl /data/third/newfile2
getfacl: Removing leading '/' from absolute path names
# file: data/third/newfile2
# owner: root
# group: root
user::rw-
group::rwx                      #effective:rw-
group:main:rwx                  #effective:rw-
mask::rw-
other::---

root@rishardkogengar:/data/main#
```

Рис. 2.7: Наследование ACL новыми файлами

19. Для проверки прав доступа выполнен вход под пользователем **carol** (член группы **third**).

    Проверка операций показала:

    - удалить файлы **newfile1** и **newfile2** пользователь **carol** не может (отказ в доступе), так как право удаления определяется правами на каталог и дополнительными ограничениями (в том числе sticky bit и отсутствием необходимых прав записи для конкретного сценария);
    - запись в файл **newfile1** запрещена, поскольку файл был создан до установки default ACL и не содержит необходимых прав записи для группы **third**;
    - запись в файл **newfile2** возможна только в случае, если у группы **third** действительно присутствует право записи в ACL данного файла (что достигает-

ся именно наследованием default ACL при создании).

```
root@rishardkogengar:/data/main#
root@rishardkogengar:/data/main# su carol
carol@rishardkogengar:/data/main$ rm /data/main/newfile1
rm: remove write-protected regular empty file '/data/main/newfile1'? y
rm: cannot remove '/data/main/newfile1': Permission denied
carol@rishardkogengar:/data/main$ rm /data/main/newfile2
rm: cannot remove '/data/main/newfile2': Permission denied
carol@rishardkogengar:/data/main$ echo "Hello world" >> /data/main/newfile1
bash: /data/main/newfile1: Permission denied
carol@rishardkogengar:/data/main$ echo "Hello world" >> /data/main/newfile2
carol@rishardkogengar:/data/main$
```

Рис. 2.8: Проверка операций

2.4 Вывод

В ходе работы были настроены и проверены: - базовые права доступа на каталоги с разграничением по группам; - специальные механизмы для общего каталога (**setgid** для наследования группы и **sticky bit** для защиты файлов пользователей друг от друга); - расширенные права доступа через ACL, включая настройку наследования прав для новых файлов через default ACL.

Результаты подтвердили, что стандартные права, специальные биты и ACL дополняют друг друга и позволяют гибко управлять доступом групп пользователей к общим ресурсам.

3 Контрольные вопросы

1. Как следует использовать команду chown, чтобы установить владельца группы для файла? Приведите пример.

Для установки владельца группы используется команда `chown` с указанием группы после двоеточия.
При этом пользователь может быть опущен.

Пример:
`chown :main file.txt` — назначает файлу **file.txt** группу **main**, не изменяя владельца файла.

Также возможно одновременно изменить владельца и группу:
`chown alice:main file.txt`

2. С помощью какой команды можно найти все файлы, принадлежащие конкретному пользователю? Приведите пример.

Для поиска файлов, принадлежащих определённому пользователю, используется команда `find` с параметром `-user`.

Пример:
`find / -user bob` — поиск всех файлов пользователя **bob** по всей файловой системе.
`find /home -user alice` — поиск файлов пользователя **alice** в каталоге `/home`.

3. Как применить разрешения на чтение, запись и выполнение для всех файлов в каталоге /data для пользователей и владельцев групп, не устанавливая никаких прав для других? Приведите пример.

Для этого используется команда `chmod` с рекурсивным применением прав и

режимом 770.

Пример:

```
chmod -R 770 /data
```

В результате: - владелец и группа получают права чтения, записи и выполнения; - для остальных пользователей доступ полностью запрещён.

4. Какая команда позволяет добавить разрешение на выполнение для файла, который необходимо сделать исполняемым?

Для добавления разрешения на выполнение используется команда chmod с символьной формой.

Пример:

`chmod +x script.sh` — добавляет право выполнения для всех категорий пользователей.

`chmod u+x script.sh` — добавляет право выполнения только владельцу файла.

5. Какая команда позволяет убедиться, что групповые разрешения для всех новых файлов, создаваемых в каталоге, будут присвоены владельцу группы этого каталога? Приведите пример.

Для этого используется установка бита идентификатора группы (**setgid**) на каталог.

Пример:

```
chmod g+s /data/main
```

После этого все новые файлы и каталоги в **/data/main** будут автоматически наследовать группу-владельца каталога.

6. Необходимо, чтобы пользователи могли удалять только те файлы, владельцами которых они являются, или которые находятся в каталоге, владельцами которого они являются. С помощью какой команды можно это сделать? Приведите пример.

Для реализации такого поведения используется **sticky bit**.

Пример:

```
chmod +t /data/main
```

Sticky bit предотвращает удаление или переименование файлов пользователями, которые не являются владельцами этих файлов, даже если у них есть права записи в каталог.

7. Какая команда добавляет ACL, который предоставляет членам группы права доступа на чтение для всех существующих файлов в текущем каталоге?

Для этого используется команда `setfacl` с указанием группы и прав доступа.

Пример:

```
setfacl -m g:third:r *
```

Команда добавляет ACL, предоставляющий группе **third** право чтения для всех файлов в текущем каталоге.

8. Что нужно сделать для гарантии того, что члены группы получат разрешения на чтение для всех файлов в текущем каталоге и во всех его подкаталогах, а также для всех файлов, которые будут созданы в этом каталоге в будущем? Приведите пример.

Необходимо: 1. Установить ACL для существующих файлов и каталогов. 2. Установить ACL по умолчанию (default ACL) для каталога.

Пример:

`setfacl -R -m g:third:r /data/main` — для всех существующих файлов и подкаталогов.

`setfacl -m d:g:third:r /data/main` — для всех файлов, создаваемых в будущем.

9. Какое значение umask нужно установить, чтобы «другие» пользователи не получали какие-либо разрешения на новые файлы? Приведите пример.

Для запрета любых прав для категории **other** используется значение `007`.

Пример:

```
umask 007
```

При таком значении: - владелец и группа получают стандартные права; - остальные пользователи не получают никаких разрешений.

10. Какая команда гарантирует, что никто не сможет удалить файл myfile

случайно?

Для защиты файла от удаления используется установка атрибута неизменяемости (**immutable**).

Пример:

```
chattr +i myfile
```

После этого файл нельзя удалить, переименовать или изменить даже пользователю **root**, пока атрибут не будет снят командой `chattr -i myfile`.