Отчёт по лабораторной работе №3

Настройка прав доступа

Лабси Мохаммед

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

1. Выполнен вход под суперпользователем с помощью команды **su -**. Получен доступ к управлению системными каталогами.

2. В корневом каталоге созданы каталоги **/data/main** и **/data/third** с использованием команды **mkdir -p /data/main /data/third**. Проверка владельцев с помощью **ls -Al /data** показала, что владельцем и группой обоих каталогов является **root**.

3. Для каталогов изменена группа-владелец: **chgrp main /data/main**, **chgrp third /data/third**. Повторная проверка через **ls -Al /data** подтвердила корректное назначение групп.

4. Установлены права доступа **770** для обоих каталогов с помощью команд **chmod 770 /data/main** и **chmod 770 /data/third**. Это позволило владельцу и группе выполнять операции чтения, записи и выполнения, полностью запретив доступ всем остальным пользователям.

5. В другом терминале выполнен вход под пользователем **bob** (**su - bob**). Пользователь успешно перешёл в каталог **/data/main** и создал файл **emptyfile** с помощью команды **touch emptyfile**, так как является членом группы **main**.

6. При попытке пользователя **bob** перейти в каталог **/data/third** и создать в нём файл было получено сообщение об отказе в доступе. Это объясняется

тем, что пользователь **bob** не состоит в группе **third**, а доступ для остальных пользователей запрещён.

2.2 Управление специальными разрешениями (setgid и sticky bit)

1. В новом терминале выполнен вход под пользователем **alice**. В каталоге **/data/main** созданы файлы **alice1** и **alice2** с помощью команды **touch**.

2. В другом терминале выполнен вход под пользователем **bob**, который также является членом группы **main**. После выполнения **ls -l** пользователь **bob** увидел файлы, созданные **alice**, и успешно удалил их командой **rm -f alice***, так как sticky bit ещё не был установлен.

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

4. Под пользователем **root** для каталога **/data/main** установлен бит идентификатора группы и sticky bit с помощью команды **chmod g+s,o+t /data/main**.

5. Под пользователем **alice** созданы файлы **alice3** и **alice4**. Проверка через **ls -l** показала, что новые файлы автоматически принадлежат группе **main**, что подтверждает корректную работу бита **setgid**.

6. Попытка пользователя **alice** удалить файлы **bob1** и **bob2** завершилась ошибкой **Operation not permitted**. Это произошло из-за установленного sticky bit, который запрещает удаление файлов пользователями, не являющимися их владельцами.

```
bob@mlabsi:/data/main$
bob@mlabsi:/data/main$ su alice
Password:
alice@mlabsi:/data/main$ touch alice1
alice@mlabsi:/data/main$ touch alice2
alice@mlabsi:/data/main$
exit
bob@mlabsi:/data/main$ ls -l
total 0
-rw-r--r--. 1 alice alice 0 Dec 12 14:10 alice1
-rw-r--r--. 1 alice alice 0 Dec 12 14:10 alice2
-rw-r--r--. 1 bob   bob   0 Dec 12 14:07 emptyfile
bob@mlabsi:/data/main$ rm -f alice*
bob@mlabsi:/data/main$ touch bob1
bob@mlabsi:/data/main$ touch bob2
bob@mlabsi:/data/main$ su
Password:
root@mlabsi:/data/main# chmod g+s,o+t /data/main/
root@mlabsi:/data/main# su alice
alice@mlabsi:/data/main$ touch alice3
alice@mlabsi:/data/main$ touch alice4
alice@mlabsi:/data/main$ ls -l
total 0
-rw-r--r--. 1 alice main 0 Dec 12 14:11 alice3
-rw-r--r--. 1 alice main 0 Dec 12 14:11 alice4
-rw-r--r--. 1 bob   bob  0 Dec 12 14:10 bob1
-rw-r--r--. 1 bob   bob  0 Dec 12 14:10 bob2
-rw-r--r--. 1 bob   bob  0 Dec 12 14:07 emptyfile
alice@mlabsi:/data/main$ rm -rf bob*
rm: cannot remove 'bob1': Operation not permitted
rm: cannot remove 'bob2': Operation not permitted
alice@mlabsi:/data/main$
```

Рис. 2.1: Проверка setgid и sticky bit в каталоге /data/main

2.3 Управление расширенными разрешениями с использованием ACL

1. Под пользователем **root** для каталога **/data/main** установлены разрешения на чтение и выполнение для группы **third**, а для каталога **/data/third** — для группы **main** с помощью команд: **setfacl -m g:third:rx /data/main setfacl -m g:main:rx /data/third**

2. С помощью команд **getfacl /data/main** и **getfacl /data/third** подтверждено, что ACL-записи применены корректно.

```
alice@mlabsi:/data/main$ su
Password:
root@mlabsi:/data/main# setfacl -m g:third:rx /data/main
root@mlabsi:/data/main# setfacl -m g:main:rx /data/third
root@mlabsi:/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@mlabsi:/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@mlabsi:/data/main#
```

Рис. 2.2: Просмотр ACL для каталогов /data/main и /data/third

3. Создан файл **newfile1** в каталоге **/data/main**. Проверка через **getfacl /data/main/newfile1** показала, что файл не унаследовал дополнительные ACL-разрешения, так как ACL по умолчанию для каталога ещё не были заданы. Аналогичный результат получен для каталога **/data/third**.

```
root@mlabsi:/data/main# touch /data/main/newfile1
root@mlabsi:/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@mlabsi:/data/main# touch /data/third/newfile1
root@mlabsi:/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@mlabsi:/data/main#
```

Рис. 2.3: Права доступа файла newfile1 без ACL по умолчанию

4. Для каталога **/data/main** установлены ACL по умолчанию для группы **third**: **setfacl -m d:g:third:rwx /data/main**. Для каталога **/data/third** добавлены ACL по умолчанию для группы **main**: **setfacl -m d:g:main:rwx /data/third**.

5. После создания файла **newfile2** в обоих каталогах проверка через **getfacl** показала, что новые файлы корректно унаследовали ACL-разрешения по умолчанию, что подтверждает правильность настройки.

```
root@mlabsi:/data/main#
root@mlabsi:/data/main# setfacl -m d:g:third:rwx /data/main
root@mlabsi:/data/main# setfacl -m d:g:main:rwx /data/third
root@mlabsi:/data/main# touch /data/main/newfile2
root@mlabsi:/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@mlabsi:/data/main# touch /data/third/newfile2
root@mlabsi:/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@mlabsi:/data/main#
```

Рис. 2.4: Наследование ACL для файлов newfile2

6. В другом терминале выполнен вход под пользователем **carol**, входящим в группу **third**. Попытки удалить файлы **newfile1** и **newfile2** в каталоге **/data/main** завершились ошибкой **Permission denied**, так как у группы **third** отсутствуют права на удаление файлов.

   Аналогично, попытки записи в файлы с помощью команды **echo “Hello, world” » /data/main/newfile1 echo “Hello, world” » /data/main/newfile2** также завершились ошибкой, поскольку для группы **third** установлены только права чтения и выполнения.

Рис. 2.5: Проверка прав доступа пользователем carol

2.4 Контрольные вопросы

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

   Для изменения владельца группы файла используется команда **chown** с указанием группы после двоеточия.

   Пример:

   chown :main myfile — устанавливает группу-владельца **main** для файла *myfile*, не изменяя владельца файла.

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

   Для поиска файлов, принадлежащих определённому пользователю, применяется команда **find** с параметром **-user**.

   Пример:

   find / -user alice — находит все файлы в системе, владельцем которых является пользователь *alice*.

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?**

   Для этого используется команда **chmod** с числовым режимом **770**.

Пример:

chmod -R 770 /data — назначает права *rwx* владельцу и группе, полностью запрещая доступ другим пользователям.

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

   Для добавления права на выполнение используется команда **chmod** с символьным режимом **+x**.

   Пример:

   chmod +x script.sh — делает файл *script.sh* исполняемым.

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

   Для этого необходимо установить бит идентификатора группы (*setgid*) на каталог с помощью команды **chmod**.

   Пример:

   chmod g+s /data/main — все новые файлы в каталоге будут наследовать группу **main**.

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?**

   Для этого используется *sticky bit*, устанавливаемый командой **chmod**.

   Пример:

   chmod +t /data/main — запрещает удаление файлов пользователями, не являющимися их владельцами.

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

Для добавления ACL используется команда **setfacl** с параметром **-m**.

Пример:

setfacl -m g:third:r * — предоставляет группе *third* права на чтение 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?**

   Необходимо рекурсивно установить ACL для существующих файлов и добавить ACL по умолчанию для каталога.

   Пример:

   setfacl -R -m g:third:r /data/main

   setfacl -m d:g:third:r /data/main

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

   Для этого необходимо установить **umask 007**.

   Пример:

   umask 007 — запрещает назначение любых прав для категории *others* при создании новых файлов.

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

    Для защиты файла от удаления можно установить атрибут неизменяемости с помощью команды **chattr**.

    Пример:

    chattr +i myfile — файл *myfile* нельзя удалить или изменить до снятия атрибута.

3 Заключение

В ходе лабораторной работы были изучены и практически применены механизмы управления правами доступа в ОС Linux. Были настроены базовые разрешения для файлов и каталогов, специальные атрибуты *setgid* и *sticky bit*, а также расширенные права доступа с использованием списков ACL. Проведённые эксперименты подтвердили корректность разграничения прав между пользователями и группами, а также показали возможности гибкой настройки доступа к ресурсам файловой системы.