

Лабораторная работа №4

Дисциплина: Информационная безопасность

Губина Ольга Вячеславовна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	10
5	Выводы	16
	Список литературы	17

Список иллюстраций

4.1	Расширенные атрибуты файла file1	10
4.2	Права, разрешающие чтение и запись для владельца файла	10
4.3	Расширенный атрибут a	10
4.4	Установка расширенного атрибута от имени суперпользователя .	11
4.5	Правильность установления атрибута	11
4.6	Дозапись в файл file1	12
4.7	Изменение имеющейся информации файла	12
4.8	Изменение имеющейся информации файла	12
4.9	Снятие расширенного атрибута a	13
4.10	Операции, неудавшиеся ранее	13
4.11	Создание нового файла	14
4.12	Присвоение расширенного атрибута i	14
4.13	Изменение имеющейся информации файла с атрибутом i	14
4.14	Удаление расширенного атрибута i	14
4.15	Операции после удаления атрибута i	15

Список таблиц

1 Цель работы

Получение практических навыков работы в консоли с расширенными атрибутами файлов.

2 Задание

- Изучить на практике действие расширенных атрибутов «а» и «і».

3 Теоретическое введение

В данной лабораторной работе нам предстоит поработать с правами доступа файлов и директорий. **Права доступа** определяют, какие действия конкретный пользователь может или не может совершать с определенными файлами и каталогами.

Есть 3 вида разрешений. Они определяют права пользователя на 3 действия: чтение, запись и выполнение. В Linux эти действия обозначаются вот так:

- **r** — read (чтение) — право просматривать содержимое файла;
- **w** — write (запись) — право изменять содержимое файла;
- **x** — execute (выполнение) — право запускать файл, если это программа или скрипт.

У каждого файла есть 3 группы пользователей, для которых можно устанавливать права доступа.

- **owner** (владелец) — отдельный человек, который владеет файлом. Обычно это тот, кто создал файл, но владельцем можно сделать и кого-то другого.
- **group** (группа) — пользователи с общими заданными правами.
- **others** (другие) — все остальные пользователи, не относящиеся к группе и не являющиеся владельцами.[1]

Чтобы увидеть текущие назначения владельца, вы можете использовать команду `ls -l`. Эта команда показывает пользователя и группу-владельца.

С помощью команды `ls` вы можете отобразить владельца файлов в данном каталоге. Иногда может оказаться полезным получить список всех файлов в системе, в которых в качестве владельца указан данный пользователь или группа. Для этого вы можете использовать `find`. Аргумент `find -user` может быть использован для этой цели.

Чтобы применить соответствующие разрешения, первое, что нужно учитывать, это владение. Для этого есть команда `chown`.^[2]

Для того, чтобы позволить обычным пользователям выполнять программы от имени суперпользователя без знания его пароля была придумана такая вещь, как SUID и SGID биты. Рассмотрим эти полномочия подробнее.

- **SUID** - если этот бит установлен, то при выполнении программы, `id` пользователя, от которого она запущена заменяется на `id` владельца файла. Фактически, это позволяет обычным пользователям запускать программы от имени суперпользователя;
- **SGID** - этот флаг работает аналогичным образом, только разница в том, что пользователь считается членом группы, с которой связан файл, а не групп, к которым он действительно принадлежит. Если SGID флаг установлен на каталог, все файлы, созданные в нем, будут связаны с группой каталога, а не пользователя. Такое поведение используется для организации общих папок;
- **Sticky-bit** - этот бит тоже используется для создания общих папок. Если он установлен, то пользователи могут только создавать, читать и выполнять файлы, но не могут удалять файлы, принадлежащие другим пользователям.^[3]

`chattr` изменяет атрибуты файлов в файловой системе Linux.

Формат символьного режима: `+-=[aAcCdDeFiJmPsStTux]`.

Оператор «+» вызывает добавление выбранных атрибутов к существующим атрибутам файлов; «-» заставляет их удалить; и «=» делает их единственными атрибутами файлов.

Буквы «aAcCdDeFijmPsStTux» выбирают **новые атрибуты для файлов**[4]:

- только добавление (a),
- без обновлений времени (A),
- сжатие (c),
- без копирования при записи (C),
- без дампа (d),
- синхронные обновления каталогов (D),
- формат экстента (e),
- поиск в каталогах без учёта регистра (F),
- неизменяемый (i),
- ведение журнала данных (j),
- без сжатия (m),
- иерархия проекта (P),
- безопасное удаление (s),
- синхронные обновления (S),
- без слияния хвостов (t),
- вершина иерархии каталогов (T),
- возможность восстановления после удаления (u)
- прямой доступ к файлам (x).

4 Выполнение лабораторной работы

1. От имени пользователя `guest` определите расширенные атрибуты файла `/home/guest/dir1/file1` командой `lsattr /home/guest/dir1/file1` (рис. 4.1).

```
[guest@ovgubina ovgubina]$ lsattr /home/guest/dir1/file1
----- /home/guest/dir1/file1
[guest@ovgubina ovgubina]$
```

Рис. 4.1: Расширенные атрибуты файла `file1`

2. Установите командой `chmod 600 file1` на файл **file1** права, разрешающие чтение и запись для владельца файла (рис. 4.2).

```
[guest@ovgubina ovgubina]$ cd /home/guest/dir1
[guest@ovgubina dir1]$ chmod 600 file1
[guest@ovgubina dir1]$
```

Рис. 4.2: Права, разрешающие чтение и запись для владельца файла

3. Попробуйте установить на файл `/home/guest/dir1/file1` расширенный атрибут `a` от имени пользователя **guest** (рис. 4.3):

```
chattr +a /home/guest/dir1/file1
```

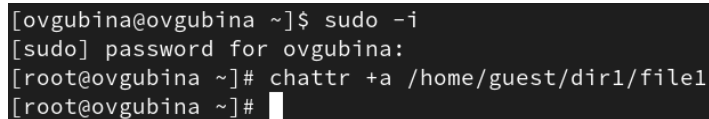
```
[guest@ovgubina dir1]$ chattr +a /home/guest/dir1/file1
chattr: Operation not permitted while setting flags on /home/guest/dir1/file1
[guest@ovgubina dir1]$
```

Рис. 4.3: Расширенный атрибут `a`

В ответ мы получили отказ от выполнения операции.

4. Зайдем в консоль с правами администратора (рис. 4.4). Попробуйте установить расширенный атрибут `a` на файл `/home/guest/dir1/file1` от имени суперпользователя (рис. 4.4):

```
chattr +a /home/guest/dir1/file1
```

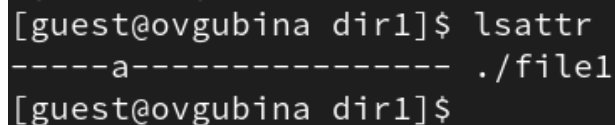


```
[ovgubina@ovgubina ~]$ sudo -i
[sudo] password for ovgubina:
[root@ovgubina ~]# chattr +a /home/guest/dir1/file1
[root@ovgubina ~]#
```

Рис. 4.4: Установка расширенного атрибута от имени суперпользователя

Установка разрешена.

5. От пользователя **guest** проверьте правильность установления атрибута:
`lsattr /home/guest/dir1/file1` (рис. 4.5)



```
[guest@ovgubina dir1]$ lsattr
-----a----- ./file1
[guest@ovgubina dir1]$
```

Рис. 4.5: Правильность установления атрибута

6. Выполните дозапись в файл **file1** слова «test» командой `echo "test" >> /home/guest/dir1/file1` (рис. 4.6).

После этого выполните чтение файла **file1** командой `cat /home/guest/dir1/file1` (рис. 4.6). Убедитесь, что слово `test` было успешно записано в `file1` командой `cat file1`.

```
[guest@ovgubina dir1]$ echo "test" >> /home/guest/dir1/file1
[guest@ovgubina dir1]$ cat file1
test020
test
[guest@ovgubina dir1]$
```

Рис. 4.6: Дозапись в файл file1

Мы можем это сделать, при просмотре файла видно, что дозапись была произведена успешно.

Дозапись возможна потому, что атрибут `a` позволяет добавлять что-либо в файл, но менять уже имеющуюся информацию с таким атрибутом нельзя.

7. Попробуйте удалить файл **file1** `rm file1` либо стереть имеющуюся в нём информацию командой `echo "abcd" > /home/guest/dir1/file1` (рис. 4.7). Попробуйте переименовать файл `mv file1 file` (рис. 4.7).

```
[guest@ovgubina dir1]$ rm file1
rm: cannot remove 'file1': Operation not permitted
[guest@ovgubina dir1]$ echo 'abcd' > file1
bash: file1: Operation not permitted
[guest@ovgubina dir1]$ mv file1 file
mv: cannot move 'file1' to 'file': Operation not permitted
[guest@ovgubina dir1]$
```

Рис. 4.7: Изменение имеющейся информации файла

Видим, что мы не сможем сделать ничего из вышеперечисленного, поскольку атрибут не позволяет нам изменять имеющуюся информацию о файле.

8. Попробуйте с помощью команды `chmod 000 file1` установить на файл **file1** права, запрещающие чтение, выполнение и запись для владельца файла (рис. 4.8). Удалось ли вам успешно выполнить указанные команды?

```
[guest@ovgubina dir1]$ chmod 000 file1
chmod: changing permissions of 'file1': Operation not permitted
[guest@ovgubina dir1]$
```

Рис. 4.8: Изменение имеющейся информации файла

Мы не можем поменять разрешения файла с данным атрибутом.

9. Снимите расширенный атрибут `a` с файла `/home/guest/dirl/file1` от имени суперпользователя командой `chattr -a /home/guest/dirl/file1` (рис. 4.9). После этого выполним операции, неудавшиеся ранее (рис. 4.10).

```
[root@ovgubina ~]# chattr -a /home/guest/dirl/file1
```

Рис. 4.9: Снятие расширенного атрибута `a`

```
[guest@ovgubina dirl]$ lsattr
----- ./file1
[guest@ovgubina dirl]$ echo "test" >> /home/guest/dirl/file1
[guest@ovgubina dirl]$ cat file1
test020
test
test
[guest@ovgubina dirl]$ echo 'abcd' > file1
[guest@ovgubina dirl]$ cat file1
abcd
[guest@ovgubina dirl]$ mv file1 file
[guest@ovgubina dirl]$ rm file
[guest@ovgubina dirl]$
```

Рис. 4.10: Операции, неудавшиеся ранее

Видим, что теперь мы можем произвести дозапись, запись, переименование и удаление файла.

Разрешения файла поменять возможно.

10. Повторите ваши действия по шагам, заменив атрибут «`a`» атрибутом «`i`». Удалось ли вам дозаписать информацию в файл? Ваши наблюдения занесите в отчёт. В результате выполнения работы вы повысили свои навыки использования интерфейса командной строки (CLI), познакомились на примерах с тем, как используются основные и расширенные атрибуты при разграничении доступа. Имели возможность связать теорию дискреционного разделения доступа (дискреционная политика безопасности) с её реализацией на практике в ОС Linux. Составили наглядные таблицы, поясняющие какие операции возможны при тех или иных установленных правах. Опробовали действие на практике расширенных атрибутов «`a`» и «`i`».

Для этого сперва создадим файл, удаленный ранее - **file**, и запишем в него какую-то информацию, для проверки (рис. 4.11).

```
[guest@ovgubina dir1]$ touch file
[guest@ovgubina dir1]$ echo 'file for test' > file
[guest@ovgubina dir1]$ cat file
file for test
[guest@ovgubina dir1]$
```

Рис. 4.11: Создание нового файла

После этого присвоим файлу расширенный атрибут **i**, который делает файл полностью неизменяемым (рис. 4.12).

```
[root@ovgubina ~]# chattr +i /home/guest/dir1/file
[root@ovgubina ~]#
```

Рис. 4.12: Присвоение расширенного атрибута i

Заново проделаем вышеперечисленные команды (рис. 4.13).

```
[guest@ovgubina dir1]$ lsattr
----i----- ./file
[guest@ovgubina dir1]$ echo "test" >> /home/guest/dir1/file
bash: /home/guest/dir1/file: Operation not permitted
[guest@ovgubina dir1]$ cat file
file for test
[guest@ovgubina dir1]$ echo 'abcd' > file
bash: file: Operation not permitted
[guest@ovgubina dir1]$ mv file file1
mv: cannot move 'file' to 'file1': Operation not permitted
[guest@ovgubina dir1]$ rm file
rm: cannot remove 'file': Operation not permitted
[guest@ovgubina dir1]$
```

Рис. 4.13: Изменение имеющейся информации файла с атрибутом i

Видим, что мы не можем произвести ни одно из выполненных действий, а именно: дозапись в файл, запись в файл, изменение имени файла, удаление.

Удалим атрибут **i** из списка расширенных атрибутов файла (рис. 4.14).

```
[root@ovgubina ~]# chattr -i /home/guest/dir1/file
[root@ovgubina ~]#
```

Рис. 4.14: Удаление расширенного атрибута i

И снова прделаем операции (рис. 4.15).

```
[guest@ovgubina dir1]$ lsattr
----- ./file
[guest@ovgubina dir1]$ echo "test" >> /home/guest/dir1/file
[guest@ovgubina dir1]$ cat file
file for test
test
[guest@ovgubina dir1]$ echo 'abcd' > file
[guest@ovgubina dir1]$ cat file
abcd
[guest@ovgubina dir1]$ mv file file1
[guest@ovgubina dir1]$ rm file1
[guest@ovgubina dir1]$ ls
[guest@ovgubina dir1]$
```

Рис. 4.15: Операции после удаления атрибута i

Видим, что теперь мы можем успешно осуществить все команды, которые не удалоаь осуществить ранее.

5 Выводы

В результате выполнения работы повысила свои навыки использования интерфейса командой строки (CLI), познакомились на примерах с тем, как используются расширенные атрибуты. Опробовала действие на практике расширенных атрибутов «a» и «i».

Список литературы

1. Права доступа в Linux [Электронный ресурс]. 2023. URL: <https://codechick.io/tutorials/unix-linux/unix-linux-permissions>.
2. Права в Linux (chown, chmod, SUID, GUID, sticky bit, ACL, umask) [Электронный ресурс]. 2023. URL: <https://habr.com/ru/articles/469667/>.
3. Права доступа к файлам в Linux [Электронный ресурс]. 2023. URL: <https://losst.pro/prava-dostupa-k-fajlam-v-linux>.
4. Атрибуты файлов в Linux [Электронный ресурс]. 2023. URL: <https://zalinux.ru/?p=6440>.