Отчет по лабораторной работе 4

Дискреционное разграничение прав в Linux. Расширенные атрибуты

Шалыгин Георгий Эдуардович

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Получение практических навыков работы в консоли с расширенными атрибутами файлов.

2 Теоретическое введение

В Linux, как и в любой многопользовательской системе, абсолютно естественным образом возникает задача разграничения доступа субъектов — пользователей к объектам — файлам дерева каталогов.

Один из подходов к разграничению доступа — так называемый дискреционный (от англ, discretion — чье-либо усмотрение) — предполагает назначение владельцев объектов, которые по собственному усмотрению определяют права доступа субъектов (других пользователей) к объектам (файлам), которыми владеют.

Дискреционные механизмы разграничения доступа используются для разграничения прав доступа процессов как обычных пользователей, так и для ограничения прав системных программ в (например, служб операционной системы), которые работают от лица псевдопользовательских учетных записей.

В Linux у каждого файла и каждого каталога есть два владельца: пользователь и группа.

Эти владельцы устанавливаются при создании файла или каталога. Пользователь, который создаёт файл становится владельцем этого файла, а первичная группа, в которую входит этот же пользователь, так же становится владельцем этого файла. Чтобы определить, есть ли у вас как у пользователя права доступа к файлу или каталогу, оболочка проверяет владение ими.

Это происходит в следующем порядке:

1. Оболочка проверяет, являетесь ли вы владельцем файла, к которому вы хотите получить доступ. Если вы являетесь этим владельцем, вы получаете разрешения и оболочка прекращает проверку.

2. Если вы не являетесь владельцем файла, оболочка проверит, являетесь ли вы участником группы, у которой есть разрешения на этот файл. Если вы являетесь участником этой группы, вы получаете доступ к файлу с разрешениями, которые для группы установлены, и оболочка прекратит проверку.
3. Если вы не являетесь ни пользователем, ни владельцем группы, вы получаете права других пользователей (Other).

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

Подробнее в [1].

2.0.1 Изменение владельца

Чтобы применить соответствующие разрешения, первое, что нужно учитывать, это владение. Для этого есть команда **chown**. Синтаксис этой команды несложен для понимания:

```
chown кто что
```

Например, следующая команда меняет владельца каталога /home/account на пользователя linda:

```
chown linda /home/account
```

2.1 Использование chmod

Для управления правами используется команда **chmod**. При использовании **chmod** вы можете устанавливать разрешения для пользователя (user), группы (group) и других (other). Вы можете использовать эту команду в двух режимах: относительный режим и абсолютный режим. В абсолютном режиме три цифры используются для установки основных разрешений.

При настройке разрешений рассчитайте необходимое вам значение. Если вы хотите установить чтение, запись и выполнение для пользователя, чтение и выполнение для группы, а также чтение и выполнение для других в файле /somefile, то вы используете следующую команду **chmod**:

```
chmod 755 /somefile
```

Подробнее в [2].

3 Выполнение лабораторной работы

1. От имени пользователя guest определим расширенные атрибуты файла /home/guest/dir1/file1 командой lsattr /home/guest/dir1/file1 (fig. 3.1).

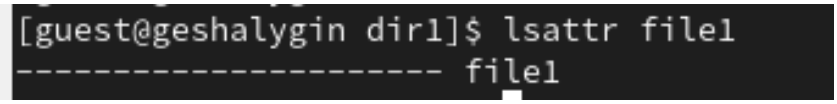

Рис. 3.1: Расширенные атрибуты

2. Установим командой chmod 600 file1 на файл file1 права, разрешающие чтение и запись для владельца файла (fig:002).

   Попробуем установить на файл /home/guest/dir1/file1 расширенный атрибут a от имени пользователя guest: chattr +a /home/guest/dir1/file1 В ответ получаем отказ от выполнения операции (fig:002).

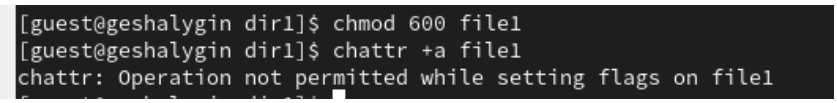

Рис. 3.2: Попытка изменения атрибутов

3. От имени супер юзера установим расширенный атрибут a на файл /home/guest/dir1/file1 (fig. 3.3).

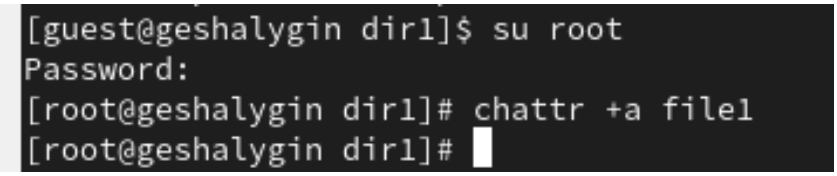

Рис. 3.3: Изменение атрибутов

4. От пользователя guest проверим правильность установления атрибута: lsattr /home/guest/dir1/file1 (fig. 3.4).

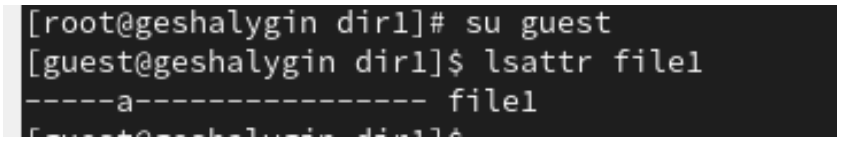

Рис. 3.4: Просмотр расширенных атрибутов

5. Выполним дозапись в файл file1 слова «test» командой echo “test” » /home/guest/dir1/file1 После этого выполните чтение файла file1 командой cat /home/guest/dir1/file. Слово test было успешно записано в file1 (fig. 3.5).

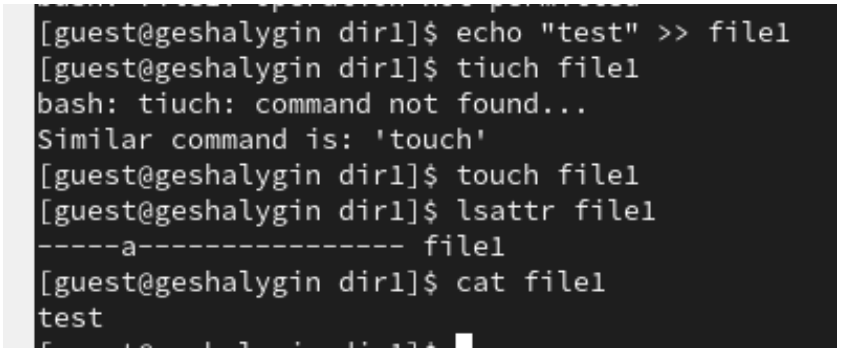

Рис. 3.5: Проверка доступа к файлу

6. Попробуем перезаписать файл file1 командой echo “abcd” > /home/guest/dirl/file1, переименовать файл. Эти действия выполнить не удается (fig. 3.6).

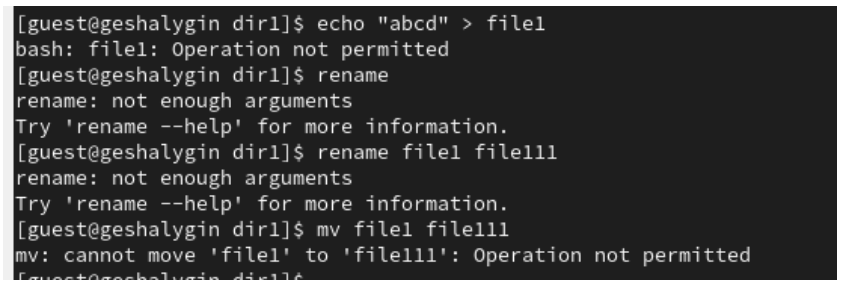

Рис. 3.6: Проверка доступа изменения

7. Также не удается понизить права файла командой chmod (fig. 3.7).

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

Рис. 3.7: Неудача

8. Снимем расширенный атрибут a с файла /home/guest/dirl/file1 от имени суперпользователя командой chattr -a /home/guest/dir1/file1 Повторим операции, которые ранее не удавалось выполнить. Теперь их выполнить можно (fig. 3.8).

```
[guest@geshalygin dir1]$ su
Password:
[root@geshalygin dir1]# chattr -a file1
[root@geshalygin dir1]# su guest
[guest@geshalygin dir1]$ echo "abcd" > file1
[guest@geshalygin dir1]$ cat file1
abcd
[guest@geshalygin dir1]$ mv file1 file111
[guest@geshalygin dir1]$ ls
file  file111  file2
```

Рис. 3.8: Проверка прав доступа для новых атрибутов

9. Повторим действия по шагам, заменив атрибут «a» атрибутом «i».

   Теперь дозапись в файл запрещена.(fig. 3.9).

```
[guest@geshalygin dir1]$ su
Password:
[root@geshalygin dir1]# chattr +i file111
[root@geshalygin dir1]# echo "qqq" > file111
bash: file111: Operation not permitted
[root@geshalygin dir1]# cat file111
abcd
[root@geshalygin dir1]# echo "qqq" >> file111
bash: file111: Operation not permitted
[root@geshalygin dir1]#
```

Рис. 3.9: Проверка доступа для новых атрибутов

4 Выводы

В результате выполнения работы мы повысили свои навыки использования интерфейса командой 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»

Список литературы

1. Кетов Д.В. Внутреннее устройство Linux. BHV, 2017. 124 с.

2. Л. М. Ухлинов. Управление доступом в ОС GNU /Linux . ОКБ САПР», Москва, Россия, 2010.