Лабораторная работа №4

Дискреционное разграничение прав в Linux. Расширенные атрибуты

Смородова Дарья Владимировна

2022 Sep 30th

Содержание

List of Tables

List of Figures

1 Цель работы

Получение практических навыков работы в консоли с расширенными атрибутами файлов.

2 Теоретическое введение

В операционной системе Linux есть много отличных функций безопасности, но она из самых важных - это система прав доступа к файлам. Linux, как последователь идеологии ядра Linux в отличие от Windows, изначально проектировался как многопользовательская система, поэтому права доступа к файлам в linux продуманы очень хорошо.

Изначально каждый файл имеет три параметра доступа:

- Чтение - разрешает получать содержимое файла, но на запись нет. Для каталога позволяет получить список файлов и каталогов, расположенных в нем;
- Запись - разрешает записывать новые данные в файл или изменять существующие, а также позволяет создавать и изменять файлы и каталоги;
- Выполнение - вы не можете выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу.

Но все эти права были бы бессмысленными, если бы применялись сразу для всех пользователей. Поэтому каждый файл имеет три категории пользователей, для которых можно устанавливать различные сочетания прав доступа:

- Владелец - набор прав для владельца файла, пользователя, который его создал или сейчас установлен его владельцем. Обычно владелец имеет все права, чтение, запись и выполнение.

- Группа - любая группа пользователей, существующая в системе и привязанная к файлу. Но это может быть только одна группа и обычно это группа владельца, хотя для файла можно назначить и другую группу.
- Остальные - все пользователи, кроме владельца и пользователей, входящих в группу файла.

Именно с помощью этих наборов полномочий устанавливаются права файлов в linux. Каждый пользователь может получить полный доступ только к файлам, владельцем которых он является или к тем, доступ к которым ему разрешен. Только пользователь Root может работать со всеми файлами независимо от их набора их полномочий. [1]

[1] Права доступа к файлам в Linux

3 Выполнение лабораторной работы [1]

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

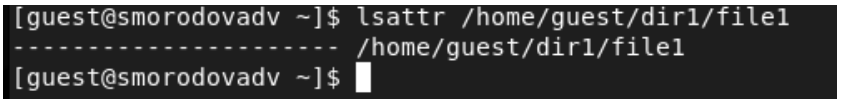

Figure 3.1: Определение расширенных атрибутов файла

2. Установим командой chmod 600 file1 на файл file1 права, разрешающие чтение и запись для владельца файла (рис. 3.2):

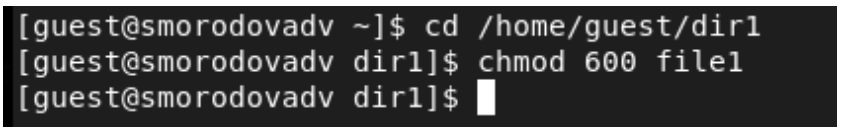

Figure 3.2: Установка прав на чтение и запись для владельца файла

3. Попробуем установить на файл /home/guest/dir1/file1 расширенный атрибут а от имени пользователя guest командой chattr +a /home/guest/dir1/file1 и получим отказ в выполнении операции (рис. 3.3):

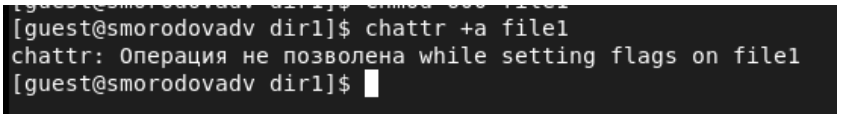

Figure 3.3: Попытка установить расширенный атрибут

[1] Методические материалы к лабораторной работе

4. Зайдем на новую консоль с правами администратора с помощью команды su. Попробуем установить расширенный атрибут a на файл /home/guest/dir1/file1 от имени суперпользователя при помощи команды chattr +a /home/guest/dir1/file1 (рис. 3.4):

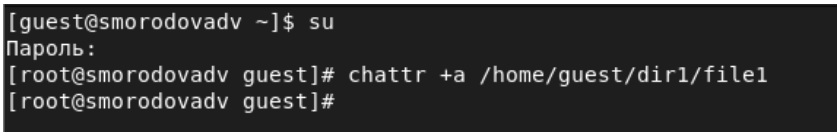

Figure 3.4: Попытка установить расширенный атрибут с правами администратора

5. От пользователя guest проверим правильность установления атрибута командой lsattr /home/guest/dir1/file1 (рис. 3.5):

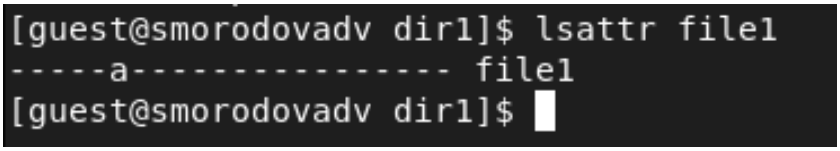

Figure 3.5: Проверка правильности установления атрибута

6. Выполним дозапись в файл file1 слова «test» командой echo “test” » /home/guest/dir1/file1, а после этого выполним чтение файла file1 командой cat /home/guest/dir1/file1 (рис. 3.6):

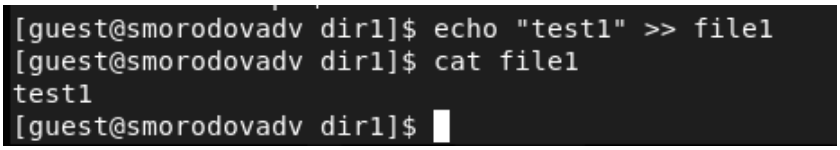

Figure 3.6: Дозапись в файл file1 слова «test»

7. Попробуем удалить файл file1 и стереть имеющуюся в нём информацию командой echo “abcd” > /home/guest/dirl/file1, затем попробуем переименовать файл. Сделать это у нас не получилось (рис. 3.7):

```
[guest@smorodovadv dir1]$ rm file1
rm: невозможно удалить 'file1': Операция не позволена
[guest@smorodovadv dir1]$ echo "abcd" >file1
bash: file1: Операция не позволена
[guest@smorodovadv dir1]$ mv file1 file2
mv: невозможно переместить 'file1' в 'file2': Операция не позволена
[guest@smorodovadv dir1]$
```

Figure 3.7: Попытка удалить файл file1 и стереть имеющуюся в нём информацию

8. Попробуем с помощью команды chmod 000 file1 установить на файл file1 права, например, запрещающие чтение и запись для владельца файла. Сделать это у нас не получилось (рис. 3.8):

```
[guest@smorodovadv dir1]$ chmod 000 file1
chmod: изменение прав доступа для 'file1': Операция не позволена
[guest@smorodovadv dir1]$
```

Figure 3.8: Попытка установить на файл file1 права

9. Снимем расширенный атрибут a с файла /home/guest/dirl/file1 от имени суперпользователя командой chattr -a /home/guest/dir1/file1 (рис. 3.9):

```
[root@smorodovadv guest]# chattr -a /home/guest/dir1/file1
[root@smorodovadv guest]#
```

Figure 3.9: Снятие расширенного атрибута

10. Повторим операции, которые ранее не удавалось выполнить. В этот раз у нас получилось это сделать (рис. 3.10):

```
[guest@smorodovadv dir1]$ echo "abcd" >file1
[guest@smorodovadv dir1]$ cat file1
abcd
[guest@smorodovadv dir1]$ mv file1 file2
[guest@smorodovadv dir1]$ ls
file  file2
[guest@smorodovadv dir1]$ chmod 000 file2
[guest@smorodovadv dir1]$ rm file2
rm: удалить защищённый от записи обычный файл 'file2'? y
[guest@smorodovadv dir1]$
```

Figure 3.10: Повтор операций

11. Повторим наши действия по шагам, заменив атрибут «a» атрибутом «i». У нас также ничего не вышло (рис. 3.11 - 3.13):

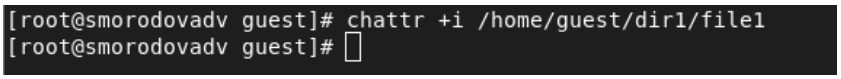

Figure 3.11: Установка расширенного атрибута «i»

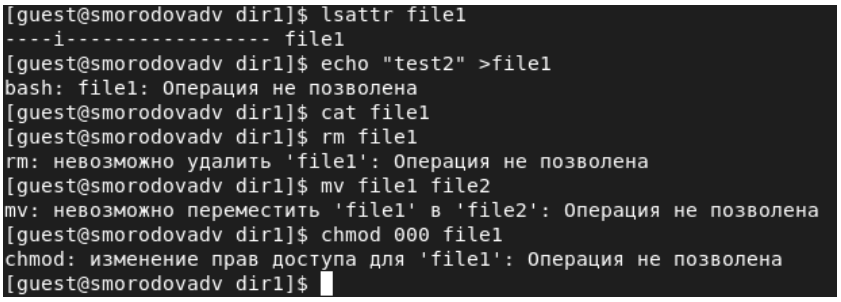

Figure 3.12: Проверка выполнения

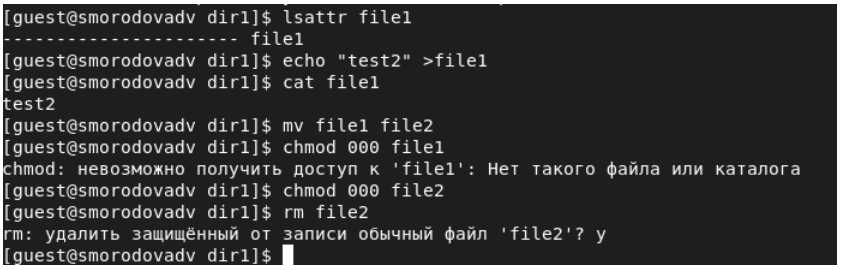

Figure 3.13: Проверка выполнения после снятия расширенного атрибута «i»

4 Выводы

В ходе данной лабораторной работы, мы повысили свои навыки использования интерфейса командой строки (CLI), познакомились на примерах с тем, как используются основные и расширенные атрибуты при разграничении доступа. Также мы имели возможность связать теорию дискреционного разделения доступа (дискреционная политика безопасности) с её реализацией на практике в ОС Linux и опробовали действие на практике расширенных атрибутов «a» и «i».

5 Список литературы

1. Методические материалы к лабораторной работе, представленные на сайте “ТУИС РУДН”

2. Права доступа к файлам в Linux