Отчёт по лабораторной работе №4

Дискреционное разграничение прав в Linux. Расширенные атрибуты

Дарья Эдуардовна Ибатулина

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Получение практических навыков работы в консоли с расширенными атрибутами файлов.

2 Задание

Снимать и устанавливать атрибуты и права доступа на файл `file1` от имени администратора и гостя. В зависимости от установленных атрибутов понять, какие действия с файлом разрешены, а какие запрещены.

3 Теоретическое введение

Права доступа определяют, какие действия конкретный пользователь может или не может совершать с определенным файлами и каталогами. С помощью разрешений можно создать надежную среду — такую, в которой никто не может поменять содержимое ваших документов или повредить системные файлы.

Расширенные атрибуты файлов Linux представляют собой пары имя:значение, которые постоянно связаны с файлами и каталогами, подобно тому как строки окружения связаны с процессом. Атрибут может быть определён или не определён. Если он определён, то его значение может быть или пустым, или не пустым.

Расширенные атрибуты дополняют обычные атрибуты, которые связаны со всеми inode в файловой системе (т. е., данные stat(2)). Часто они используются для предоставления дополнительных возможностей файловой системы, например, дополнительные возможности безопасности, такие как списки контроля доступа (ACL), могут быть реализованы через расширенные атрибуты.

Установить атрибуты:

- chattr filename

Значения:

- chattr +a # только добавление. Удаление и переименование запрещено;
- chattr +A # не фиксировать данные об обращении к файлу
- chattr +c # сжатый файл

- chattr +d # неархивируемый файл
- chattr +i # неизменяемый файл
- chattr +S # синхронное обновление
- chattr +s # безопасное удаление, (после удаления место на диске переписывается нулями)
- chattr +u # неудаляемый файл
- chattr -R # рекурсия

Просмотреть атрибуты:

- lsattr filename

Опции:

- lsattr -R # рекурсия
- lsattr -a # вывести все файлы (включая скрытые)
- lsattr -d # не выводить содержимое директории

4 Выполнение лабораторной работы

1. От имени пользователя guest определяю расширенные атрибуты файла /home/guest/dir1/file1 командой lsattr /home/guest/dir1/file1 (рис. [4.1]):

Рис. 4.1: Проверка расширенных атрибутов файла file1

2. Установлю командой chmod 600 file1 на файл file1 права, разрешающие чтение и запись для владельца файла (рис. [4.2]):

Рис. 4.2: Установка прав на чтение и запись файла file1 для владельца файла

3. Пробую установить на файл /home/guest/dir1/file1 расширенный атрибут a от имени пользователя guest: chattr +a /home/guest/dir1/file1. В ответ получаю отказ от выполнения операции (рис. [4.3]):

Рис. 4.3: Установка расширенного атрибута a на файл file1

4. Теперь проделаю то же самое, но под учётной записью администратора (рис. [4.4]):

Рис. 4.4: Установка расширенного атрибута a на файл file1 от имени администратора

5. От пользователя `guest` проверяю правильность установления атрибута при помощи команды `lsattr /home/guest/dir1/file1` (рис. [4.5]):

Рис. 4.5: Проверка правильности установленных атрибутов

6. Выполняю дозапись в файл `file1` слова «test» командой `echo >> "test" /home/guest/dir1/file1`. Также дозапишу в файл и строку “abcd”: `echo "abcd" > /home/guest/dirl/file1`. После этого выполняю чтение файла `file1` командой `cat /home/guest/dir1/file1`. Убедилась, что слова “test” и “abcd” были успешно записаны в file1 (рис. [4.6]):

Рис. 4.6: Проверка записи в файл

7. Пытаюсь удалить или переименовать файл `file1`. Но получаю отказ (рис. [4.7], [4.8]):

Рис. 4.7: Удаление файла (отказ)

Рис. 4.8: Переименование файла (отказ)

8. Устанавливаю командой `chmod 000 file1` права, запрещающие чтение и запись для владельца файла. Успешно выполнить данную команду мне не удалось (рис. [4.9]):

Рис. 4.9: Изменение прав доступа к файлу (отказ)

9. Снимаю расширенный атрибут `a` с файла `/home/guest/dirl/file1` от имени суперпользователя командой `chattr -a /home/guest/dir1/file1` (рис. [4.10]):

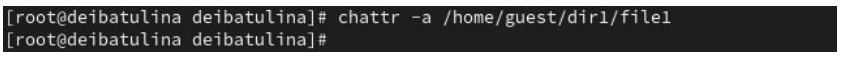

Рис. 4.10: Снятие расширенного атрибута a с файла от имени администратора

10. Повторяю операции, которые ранее не удавалось выполнить. Это переименование файла и изменение прав доступа к файлу. Удалить я бы тоже смогла, но не стала этого делать, так как в дальнейшем этот файл пригодится (рис. [4.11], [4.12]):

Рис. 4.11: Переименование файла (успешно)

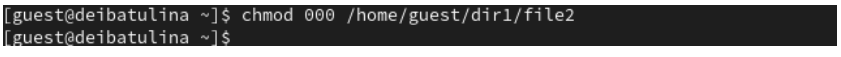

Рис. 4.12: Изменение прав доступа к файлу (успешно)

11. Повторяю свои действия по шагам, заменив атрибут «a» атрибутом «i». В случае `guest` получаю ошибку, в случае `root` команда успешно выполняется (рис. [4.13], [4.14]):

Рис. 4.13: Установка атрибута “a” на файл file1 под именем guest (отказ)

Рис. 4.14: Установка атрибута “a” на файл file1 под именем root (успех)

12. От имени пользователя `guest` переименовать, удалить, либо записать файл `file1` теперь нельзя (рис. [4.15]):

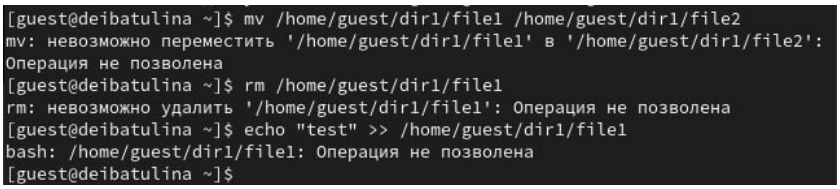

Рис. 4.15: Переименование, удаление, запись в файл file1 (отказ)

5 Выводы

В результате выполнения работы я повысила свои навыки использования интерфейса командой строки (CLI), познакомилась на примерах с тем, как используются основные и расширенные атрибуты при разграничении доступа. Имела возможность связать теорию дискреционного разделения доступа (дискреционная политика безопасности) с её реализацией на практике в ОС Linux. Опробовала действие на практике расширенных атрибутов «a» и «i».

Список литературы