Отчёт по лабораторной работе №4

Основы информационной безопасности

Бережной Иван Александрович

Содержание

Список иллюстраций

1 Цель работы

Получить практические навыки работы в консоли с расширенными атрибутами файлов.

2 Задание

1. Проверка расширенного атрибута `a`
2. Проверка расширенного атрибута `i`

3 Теоретическое введение

1. Типы прав доступа 1.1. Чтение (r): Разрешает просмотр содержимого файла или списка файлов в каталоге.

   1.2. Запись (w): Разрешает изменение содержимого файла или создание/удаление файлов в каталоге.

   1.3. Выполнение (x): Разрешает выполнение файла как программы или вход в каталог.

2. Категории пользователей

   - Владелец (Owner): Пользователь, который создал файл или каталог.
   - Группа (Group): Набор пользователей, которые имеют общие права доступа к файлу или каталогу.
   - Остальные (Others): Все остальные пользователи, не являющиеся владельцем или членом группы.

3. Представление прав доступа Права доступа отображаются в виде символов или цифр:

Символьное представление: rwxr-xr–
Первые три символа (rwx) — права владельца.
Следующие три (r-x) — права группы.
Последние три (r–) — права остальных.
Цифровое представление: Каждое право имеет числовое значение:
r = 4

w = 2

x = 1 Например, rwxr-xr– в цифровом виде: 754.

4. Команды для управления правами доступа chmod: Изменяет права доступа.

Пример: chmod 755 file.txt — устанавливает права rwxr-xr-x.

chown: Изменяет владельца файла или каталога.

Пример: chown user:group file.txt — изменяет владельца и группу.

chgrp: Изменяет группу файла или каталога.

Пример: chgrp group file.txt.

5. Особые права доступа SUID (Set User ID): Если установлен для файла, он выполняется с правами владельца.

SGID (Set Group ID): Если установлен для файла, он выполняется с правами группы. Для каталога — новые файлы наследуют группу каталога.

Sticky Bit: Если установлен для каталога, только владелец файла может удалить или переименовать файл в этом каталоге.

4 Выполнение лабораторной работы

4.1 Проверка расширенного атрибута a

Зайдём в терминал от имени пользователя guest и определим расширенные атрибуты файла dir1/file1 (рис. 4.1). Как видим, прав нет. Установим на файл file1 права на чтение, запись для владельца файла командой `chmod 600 dir1/file1` (рис. 4.2).

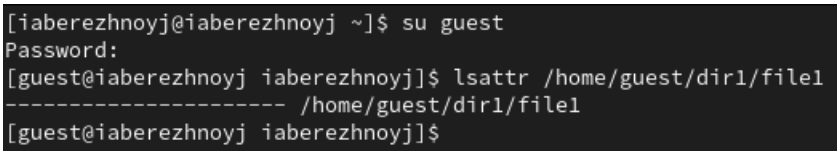

Рис. 4.1: Определение атрибутов

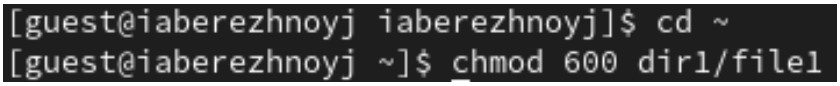

Рис. 4.2: Установка прав 600

Установим на упомянутый файл расширенный атрибут a от имени guest. Получили ошибку (рис. 4.3). Попробуем повторить действие от имени другого пользователя с правами администратора (рис. 4.4). Получилось.

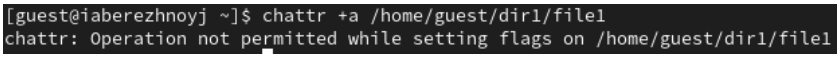

Рис. 4.3: Попытка установки расширенного атрибута

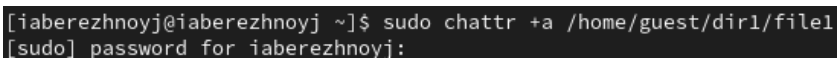

Рис. 4.4: Установка расширенного атрибута

От имени пользователя guest проверим правильность установления атрибута (рис. 4.5).

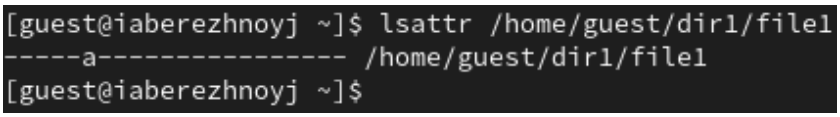

Рис. 4.5: Проверка расширенного атрибута

Выполним дозапись текста в файл file1 (рис. 4.6). Работает.

Рис. 4.6: Дозапись в файл

Попробуем стереть имеющуюся информацию в файле - неудача (рис. 4.7). Попробуем переименовать файл - также безуспешно (рис. 4.8). И, наконец, попробуем изменить права доступа к файлу. Впрочем, ничего неожиданного (рис. 4.9).

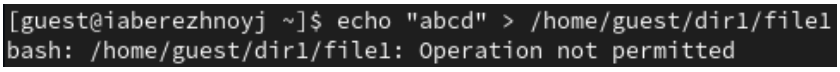

Рис. 4.7: Попытка стереть файл

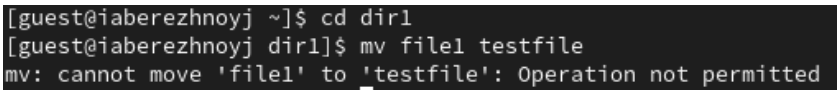

Рис. 4.8: Попытка переименовать файла

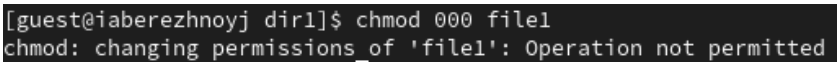

Рис. 4.9: Попытка изменения прав

Снимем расширенный атрибут с помощью администратора (рис. 4.10). Попробуем повторить неудавшиеся действия. Всё получилось (рис. 4.11).

```
[iaberezhnoyj@iaberezhnoyj ~]$ sudo chattr -a /home/guest/dir1/file1
[sudo] password for iaberezhnoyj:
[iaberezhnoyj@iaberezhnoyj ~]$
```

Рис. 4.10: Отмена атрибута a

```
[guest@iaberezhnoyj dir1]$ echo "abcd" > /home/guest/dir1/file1
[guest@iaberezhnoyj dir1]$ cat file1
abcd
[guest@iaberezhnoyj dir1]$ mv file1 test_file
[guest@iaberezhnoyj dir1]$ ls
test_file
[guest@iaberezhnoyj dir1]$ mv test_file file1
[guest@iaberezhnoyj dir1]$ chmod 000 file1
[guest@iaberezhnoyj dir1]$ ls -la
total 4
drwx------. 2 guest guest  19 Apr  5 14:48 .
drwxrwx---. 4 guest guest 151 Apr  5 14:36 ..
----------. 1 guest guest   5 Apr  5 14:47 file1
[guest@iaberezhnoyj dir1]$
```

Рис. 4.11: Повтор действий

4.2 Проверка расширенного атрибута i

Выдадим расширенный атрибут i, как это было с a (рис. 4.12) и повторим проделанные действия (рис. 4.13). Ничего не работает.

```
[iaberezhnoyj@iaberezhnoyj ~]$ sudo chattr +i /home/guest/dir1/file1
[iaberezhnoyj@iaberezhnoyj ~]$
```

Рис. 4.12: Расширенный атрибут i

```
[guest@iaberezhnoyj dir1]$ chmod 600 file1
chmod: changing permissions of 'file1': Operation not permitted
[guest@iaberezhnoyj dir1]$ lsattr /home/guest/dir1/file1
lsattr: Permission denied While reading flags on /home/guest/dir1/file1
[guest@iaberezhnoyj dir1]$ echo "text" > /home/guest/dir1/file1
bash: /home/guest/dir1/file1: Operation not permitted
[guest@iaberezhnoyj dir1]$ mv file1 test_file
mv: cannot move 'file1' to 'test_file': Operation not permitted
[guest@iaberezhnoyj dir1]$ echo "test" >> /home/guest/dir1/file1
bash: /home/guest/dir1/file1: Operation not permitted
[guest@iaberezhnoyj dir1]$
```

Рис. 4.13: Повтор действий 2

5 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».