Лабораторная работа №4

Дисциплина: Информационная безопасность

Губина Ольга Вячеслвовна

Содержание

# 1 Цель работы

Получение практических навыков работы в консоли с расширенными атрибутами файлов.

# 2 Задание

* Изучить на практике действие расширенных атрибутов «а» и «i».

# 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 определенным 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).

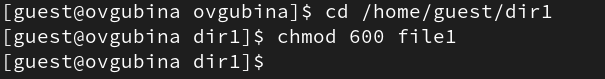
# 4 Выполнение лабораторной работы

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



Расширенные атрибуты файла file1

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



Права, разрешающие чтение и запись для владельца файла

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

chattr +a /home/guest/dir1/file1

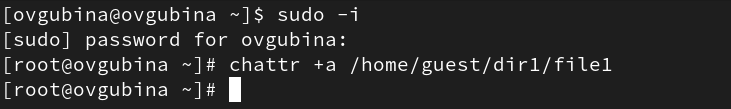


Расширенный отрибут а

В ответ мы получили отказ от выполнения операции.

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

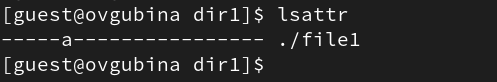
chattr +a /home/guest/dir1/file1



Установка расширенного атрибута от имени суперпользователя

Установка разрешена.

1. От пользователя **guest** проверьте правильность установления атрибута: lsattr /home/guest/dir1/file1 (рис. ??)



Правильность установления атрибута

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

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

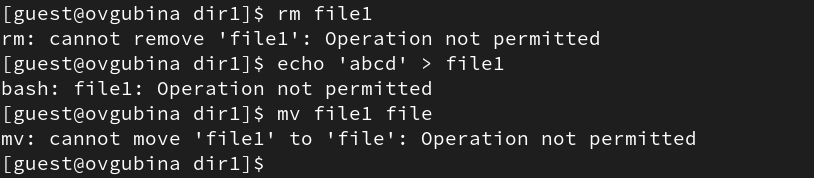


Дозапись в файл file1

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

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

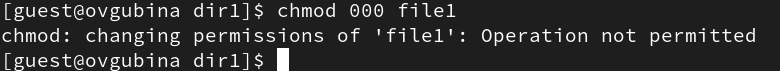
1. Попробуйте удалить файл **file1** rm file1 либо стереть имеющуюся в нём информацию командой echo "abcd" > /home/guest/dirl/file1 (рис. ??). Попробуйте переименовать файл mv file1 file (рис. ??).



Изменение имеющейся информации файла

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

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



Изменение имеющейся информации файла

Мы не можем поменять разрешения файла с данным атрибутом.

1. Снимите расширенный атрибут a с файла **/home/guest/dirl/file1** от имени суперпользователя командой chattr -a /home/guest/dir1/file1 (рис. ??). После этого выполним опрации, неудавшиеся ранее (рис. ??).

Снятие расширенного атрибута а

Снятие расширенного атрибута а



Операции, неудавшиеся ранее

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

Разрешения файла поменять возможно.

1. 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 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 «а» и «i».

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



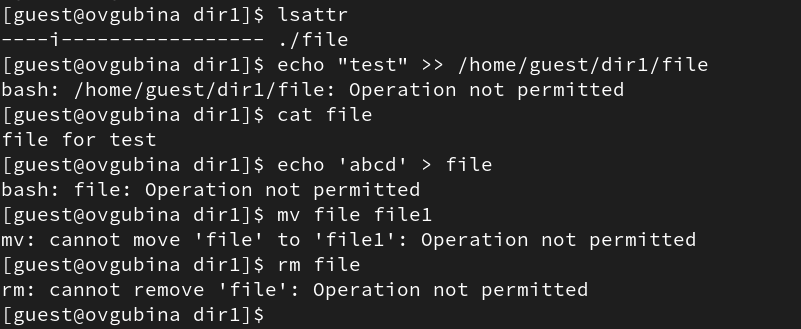
Создание нового файла

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

Присвоение расширенного атрибута i

Присвоение расширенного атрибута i

Заново проделаем вышеперечисленные команды (рис. ??).



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. ??).

Удаление расширенного атрибута i

Удаление расширенного атрибута i

И снова прделаем операции (рис. ??).



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 «а» 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>.