

Лабораторная работа №4

Дисциплина: Основы информационной безопасности

Феоктистов Владислав Сергеевич

Содержание

| | | |
|----------|---|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Теоретическое введение | 7 |
| 3.1 | Изменение атрибутов | 7 |
| 3.2 | Изменение расширенных атрибутов | 8 |
| 3.3 | Таблицы | 8 |
| 4 | Выполнение лабораторной работы | 11 |
| 4.1 | Исполнение команд в консоли | 11 |
| 5 | Выводы | 16 |
| | Список литературы | 17 |

Список иллюстраций

| | | |
|-----|--|----|
| 4.1 | Установка и просмотр прав и расширенных атрибутов | 11 |
| 4.2 | Работа с расширенным атрибутом “а” | 13 |
| 4.3 | Работа без расширенного атрибута “а” | 13 |
| 4.4 | Возвращение предыдущего имени файла и установка прав | 14 |
| 4.5 | Работа с расширенным атрибутом “і” | 14 |

Список таблиц

| | | |
|-----|---|---|
| 3.1 | Описание некоторых каталогов файловой системы GNU Linux . . | 9 |
| 3.2 | Описание некоторых используемых в работе команд | 9 |

1 Цель работы

Целью данной работы является получение практических навыков работы в консоли с расширенными атрибутами файлов.

2 Задание

- От имени пользователя *guest* проверить существующие расширенные атрибуты файла *file*, установить на файл права 600 и попытаться добавить ему атрибут “a”;
- попытаться установить расширенный атрибут “a” от имен суперпользователя;
- от пользователя *guest* проверить изменения, попробовать добавить информацию файл, считать содержимое, стереть данные, переименовать файл, установить новые права на файл;
- повторить операции, которые ранее не удалось выполнить, без расширенного атрибута “a”;
- повторить все действия с расширенным атрибутом “i”.

3 Теоретическое введение

3.1 Изменение атрибутов

В ОС Linux права доступа к файлам, атрибуты и владение управляют уровнем доступа, который система обрабатывает, а пользователи имеют к файлам. Это гарантирует, что только авторизованные пользователи и процессы могут получить доступ к определенным файлам и каталогам. Атрибуты состоят из девяти битов, которые и определяют права для разных групп пользователей. Первая тройка битов определяет права доступа для владельца, вторая тройка - для членов группы, последняя тройка - для всех остальных пользователей в системе. Каждая тройка битов (класс пользователей) определяет права на чтение, запись и исполнение. Эта концепция позволяет контролировать, какие пользователи могут читать, записывать (изменять) или выполнять файлы/каталоги.

Чтобы просмотреть права доступа к файлу, используется команда `ls` с опцией `-l`. Первый символ указывает тип файла. Это может быть обычный файл (`-`), каталог (`d`), символическая ссылка (`l`) или другие специфические типы файлов. Следующие девять символов предоставляют доступ к файлу, три тройки по три символа каждая (три пользователя, три типа прав: `r` - чтение, `w` - запись, `x` - исполнение).

Права доступа к файлу/каталогу можно изменить с помощью команды `chmod`. Только `root`, владелец файла или пользователь с привилегией `sudo` могут изменять права доступа к файлу или каталогу. Разрешения можно указывать с помощью символического, числового или справочного режимов [1].

3.2 Изменение расширенных атрибутов

Помимо битов режима файла, которые управляют разрешениями пользователей и групп на чтение, запись и выполнение, некоторые файловые системы поддерживают атрибуты файла (расширенные атрибуты), которые позволяют дополнительно настраивать допустимые операции с файлами.

Пакет `e2fsprogs` содержит программы `lsattr(1)` и `chattr(1)`, которые позволяют просмотреть и изменить атрибуты файла соответственно.

Здесь приведены некоторые полезные атрибуты. Не все файловые системы поддерживают каждый упомянутый атрибут.

- `a` - `append only`: Файл может быть открыт только для добавления.
- `c` - `compressed`: Включить сжатие на уровне файловой системы для файла.
- `i` - `immutable`: Не может быть изменён, удалён или переименован. Может быть установлен только пользователем `root`.
- `j` - `data journaling`: Использовать журнал для записи данных файла так же, как и метаданных.
- `m` - `no compression`: Отключить сжатие на уровне файловой системы для файла.
- `A` - `no atime update`: Время получения доступа к файлу не будет обновляться.
- `C` - `no copy on write`: Отключение `copy-on-write` на поддерживающих это файловых системах.

Дополнительную информацию можно получить на сайте [2].

3.3 Таблицы

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

| Имя каталога | Описание каталога |
|--------------|--|
| / | Корневая директория, содержащая всю файловую систему |
| /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям |
| /etc | Общесистемные конфигурационные файлы и файлы конфигурации установленных программ |
| /home | Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя |
| /media | Точки монтирования для сменных носителей |
| /root | Домашняя директория пользователя root |
| /tmp | Временные файлы |
| /usr | Вторичная иерархия для данных пользователя |

Таблица 3.2: Описание некоторых используемых в работе команд

| Команда | Описание команды |
|---------|--|
| cat | Вывод содержимого указанного файла. |
| ls | Выводит содержимое каталога. Опция -l выводит дополнительную информацию, -a отображает скрытые файлы, в названии которых в самом начале стоит символ '.' |
| lsattr | Просмотр атрибутов файлов/каталогов в файловой системе Linux. |
| chmod | Изменение прав доступа к файлам и каталогам, используемых в Unix-подобных операционных системах. |
| echo | Вывод переданных аргументов, строки, текста. |
| chattr | Изменяет атрибуты файлов/каталогов в файловой системе Linux. |

Ко-

манда Описание команды

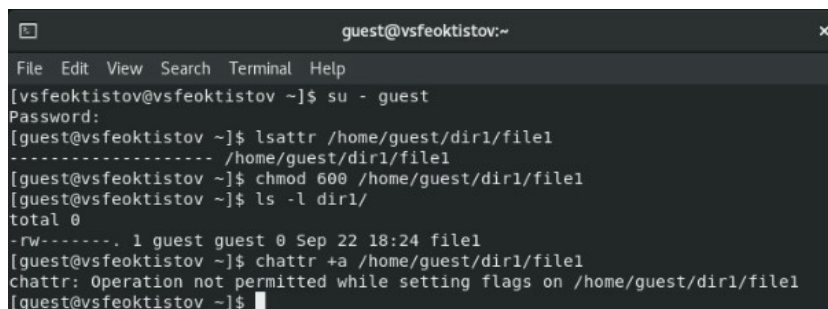
rename Переименование файла/каталога. Формат rename [старое имя] [новое имя] [путь до файла].

Более подробно об Unix см. в [3–8].

4 Выполнение лабораторной работы

4.1 Исполнение команд в консоли

Определим от имени пользователя *guest* (в случае, если Вы сейчас находитесь в системе под именем другого пользователя, то нужно будет зайти под пользователем *guest* с помощью команды `su - guest`) расширенные атрибуты файла */home/guest/dir1/file1* [**cmd:** `lsattr /home/guest/dir1/file1`] (каталог *dir1* и файл *file1* были созданы в предыдущих лабораторных работах). Из рисунка 4.1 видно, что в начале файл не имел никаких расширенных атрибутов. После установим разрешение только на чтение и запись файла *file1* для его владельца, попробуем установить для этого же файла расширенный атрибут “а” от того же пользователя [**cmd:** `chattr +a /home/guest/dir1/file1`] (рис. 4.1).



```
guest@vsfeoktistov:~  
File Edit View Search Terminal Help  
[vsfeoktistov@vsfeoktistov ~]$ su - guest  
Password:  
[guest@vsfeoktistov ~]$ lsattr /home/guest/dir1/file1  
----- /home/guest/dir1/file1  
[guest@vsfeoktistov ~]$ chmod 600 /home/guest/dir1/file1  
[guest@vsfeoktistov ~]$ ls -l dir1/  
total 0  
-rw-----. 1 guest guest 0 Sep 22 18:24 file1  
[guest@vsfeoktistov ~]$ chattr +a /home/guest/dir1/file1  
chattr: Operation not permitted while setting flags on /home/guest/dir1/file1  
[guest@vsfeoktistov ~]$
```

Рис. 4.1: Установка и просмотр прав и расширенных атрибутов

Как видно из рисунка 4.1, у пользователя *guest* не достаточно прав для установление расширенных атрибутов (Operation not permitted).

Попробуем установить расширенные атрибуты от имени суперпользователя. Для этого введем команду `su` и повторим команду [**cmd:** `chattr +a`

/home/guest/dir1/file1]. Из рисунка 4.2 видно, что теперь эта команда выполнена без ошибок. Проверим, что атрибуты действительно изменились, для этого выполним команду `lsattr` от имени пользователя *guest* [**cmds:** `su - guest` и `lsattr /home/guest/dir1/file1`]. Из того же рисунка видно, что теперь у файла *file1* появился атрибут “a”.

Далее попытаемся выполнить ряд действий над файлом. Попробуем дозаписать в файл *file1* слово “test” командой `echo “test” » /home/guest/dir1/file1`, а после считать информацию с файла, чтобы убедиться, что слово “test” было успешно добавлено в файл [**cmd:** `cat /home/guest/dir1/file1`]. Для того, чтобы убедиться, что сообщения добавляется именно в конец файла (не стирает предыдущую информацию, чтобы записать новую), можно повторить недавние команды. После попытаемся стереть и перезаписать информацию в файле *file1* [**cmd:** `echo “abcd” > cat /home/guest/dir1/file1`]. Появится сообщение о том, что такое действие над файлом не возможно, поскольку атрибут “a” разрешает только дополнять данные, а не перезаписывать их (командой `cat` можно проверить, что содержимое файла действительно не изменилось). Также не будет доступна команда переименования файла [**cmd:** `rename file1 file2 /home/guest/dir1/file1`] и команда переопределения прав доступа [**cmd:** `chmod 000 dir1/file1`], поскольку они приводят к изменению метаданных файла (эти действия можно проверить с помощью команд `ls` и `ls -l`) (рис. 4.2).

```
guest@vsfeoktistov:~  
File Edit View Search Terminal Help  
[guest@vsfeoktistov ~]$ chattr +a /home/guest/dir1/file1  
chattr: Operation not permitted while setting flags on /home/guest/dir1/file1  
[guest@vsfeoktistov ~]$ su  
Password:  
[root@vsfeoktistov guest]# chattr +a /home/guest/dir1/file1  
[root@vsfeoktistov guest]# su - guest  
[guest@vsfeoktistov ~]$ lsattr /home/guest/dir1/file1  
-----a----- /home/guest/dir1/file1  
[guest@vsfeoktistov ~]$ echo "test" >> /home/guest/dir1/file1  
[guest@vsfeoktistov ~]$ cat /home/guest/dir1/file1  
test  
[guest@vsfeoktistov ~]$ echo "test" >> /home/guest/dir1/file1  
[guest@vsfeoktistov ~]$ cat /home/guest/dir1/file1  
test  
test  
[guest@vsfeoktistov ~]$ echo "abcd" > /home/guest/dir1/file1  
-bash: /home/guest/dir1/file1: Operation not permitted  
[guest@vsfeoktistov ~]$ rename file1 file2 dir1/file1  
rename: dir1/file1: rename to dir1/file2 failed: Operation not permitted  
[guest@vsfeoktistov ~]$ chmod 000 file1  
chmod: cannot access 'file1': No such file or directory  
[guest@vsfeoktistov ~]$ chmod 000 /home/guest/dir1/file1  
chmod: changing permissions of '/home/guest/dir1/file1': Operation not permitted  
[guest@vsfeoktistov ~]$
```

Рис. 4.2: Работа с расширенным атрибутом “а”

Теперь же попробуем повторить операции, которые выдали сообщение о невозможности исполнения, только уже без расширенного атрибута “а”. Для снятия атрибута “а” используем команду `chattr -a /home/guest/dir1/file1` от имени суперпользователя (рис. 4.3).

```
guest@vsfeoktistov:~  
File Edit View Search Terminal Help  
[guest@vsfeoktistov ~]$ su  
Password:  
[root@vsfeoktistov guest]# chattr -a /home/guest/dir1/file1  
[root@vsfeoktistov guest]# su - guest  
[guest@vsfeoktistov ~]$ lsattr /home/guest/dir1/file1  
----- /home/guest/dir1/file1  
[guest@vsfeoktistov ~]$ cat /home/guest/dir1/file1  
test  
test  
[guest@vsfeoktistov ~]$ echo "abcd" > /home/guest/dir1/file1  
[guest@vsfeoktistov ~]$ cat /home/guest/dir1/file1  
abcd  
[guest@vsfeoktistov ~]$ rename file1 file2 dir1/file1  
[guest@vsfeoktistov ~]$ ls dir1/  
file2  
[guest@vsfeoktistov ~]$ chmod 000 /home/guest/dir1/file2  
[guest@vsfeoktistov ~]$ ls -l dir1/  
total 4  
-----  
1 guest guest 5 Sep 22 18:40 file2  
[guest@vsfeoktistov ~]$
```

Рис. 4.3: Работа без расширенного атрибута “а”

По итогу, все эти операции выполнились без ошибок, поскольку теперь ограничение на только добавление информации исчезло.

Так как права и имя файла *file1* было изменено на *file2*, то для повторения предыдущих действий с расширенным атрибутом “i” необходимо будет вернуть права и предыдущее название [**cmds:** *rename file2 file1 /home/guest/dir1/file2* и *chmod 700 /home/guest/dir1/file2*]. Для проверки изменений используются команды *ls* и *ls -l* (рис. 4.4).

```

guest@vsfeoktistov:~$ rename file2 file1 dir1/file2
guest@vsfeoktistov ~]$ ls dir1/
file1
guest@vsfeoktistov ~]$ chmod 700 dir1/file1
guest@vsfeoktistov ~]$ ls -l dir1/
total 4
-rwx-----. 1 guest guest 5 Sep 22 18:40 file1
guest@vsfeoktistov ~]$

```

Рис. 4.4: Возвращение предыдущего имени файла и установка прав

Повторим действия по шагам, заменив атрибут “a” атрибутом “i”. Так как атрибут “a” был ранее уже снят, то необходимо только добавить атрибут “i” от имени суперпользователя [**cmd:** *chattr +a /home/guest/dir1/file1*] (рис. 4.5).

```

guest@vsfeoktistov:~$ su
Password:
[root@vsfeoktistov guest]# chattr +i /home/guest/dir1/file1
[root@vsfeoktistov guest]# su - guest
guest@vsfeoktistov ~]$ lsattr /home/guest/dir1/file1
----i----- /home/guest/dir1/file1
guest@vsfeoktistov ~]$ cat /home/guest/dir1/file1
abcd
guest@vsfeoktistov ~]$ echo "test" >> /home/guest/dir1/file1
-bash: /home/guest/dir1/file1: Operation not permitted
guest@vsfeoktistov ~]$ cat /home/guest/dir1/file1
abcd
guest@vsfeoktistov ~]$ echo "abcde" > /home/guest/dir1/file1
-bash: /home/guest/dir1/file1: Operation not permitted
guest@vsfeoktistov ~]$ cat /home/guest/dir1/file1
abcd
guest@vsfeoktistov ~]$ rename file1 file2 dir1/file1
rename: dir1/file1: rename to dir1/file2 failed: Operation not permitted
guest@vsfeoktistov ~]$ ls dir1/
file1
guest@vsfeoktistov ~]$ chmod 000 /home/guest/dir1/file1
chmod: changing permissions of '/home/guest/dir1/file1': Operation not permitted
guest@vsfeoktistov ~]$ ls -l dir1/
total 4
-rwx-----. 1 guest guest 5 Sep 22 18:40 file1
guest@vsfeoktistov ~]$

```

Рис. 4.5: Работа с расширенным атрибутом “i”

Так как расширенный атрибут “i” делает файл неизменяемым, то любое действие, которое будет приводить к изменению файла, будет отклонено. Т.е. добавле-

ние, перезапись информации в файле *file1*, а также переименование и изменение прав файла будет невозможно (рис. 4.5).

5 Выводы

В процессе выполнения лабораторной работы приобрел практические навыки работы в консоли с расширенными атрибутами файлов через терминал; на примерах понял, как используются расширенные атрибуты “a” и “i” при разграничении доступа.

Список литературы

1. Понимание прав доступа к файлам в Linux [Электронный ресурс]. Baks, 2021. URL: <https://baks.dev/article/terminal/understanding-linux-file-permissions?ysclid=l8czjs1hnp553393513>.
2. Атрибуты файла [Электронный ресурс]. Archlinux, 2022. URL: [https://wiki.archlinux.org/title/File_permissions_and_attributes_\(%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9\)](https://wiki.archlinux.org/title/File_permissions_and_attributes_(%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9)).
3. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: <https://www.gnu.org/software/bash/manual/>.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
5. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
6. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
7. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
8. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.