Отчёт по лабораторной работе №5

дисциплина: Информационная безопасность

Быстров Глеб Андреевич

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

В данной лабораторной работе мне будет необходимо изучить механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получить практические навыки работы в консоли с дополнительными атрибутами. Рассмотреть работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

2 Задание

Последовательно выполнить пункты в терминале Linux.

3 Теоретическое введение

В Linux у каждого файла и каждого каталога есть два владельца: пользователь и группа.

Эти владельцы устанавливаются при создании файла или каталога. Пользователь, который создаёт файл становится владельцем этого файла, а первичная группа, в которую входит этот же пользователь, так же становится владельцем этого файла. Чтобы определить, есть ли у вас как у пользователя права доступа к файлу или каталогу, оболочка проверяет владение ими. [1].

Выделяют три категории пользователей, которым могут предоставляться права на файл:

- Сам владелец (u – user) объекта – конкретный пользователь, чье имя числится в атрибутах файла как имя владельца этого файла. Обычно если пользователь создает файл, то он автоматически записывается как его владелец.
- Группа (g – group), к которой принадлежит владелец файла. Когда в Linux создается пользователь, то для него создается одноименная группа. Однако средствами администрирования системы можно объединять пользователей в различные группы. При этом конкретный пользователь может входить в состав нескольких групп. Группы позволяют предоставлять права доступа к ресурсам сразу нескольким людям, но при этом ограниченному кругу лиц.
- Все остальные (o – other) – это все те, кто не является владельцем файла и не принадлежит к группе владельца файла. То есть любой посторонний пользователь.

Чтение, запись, выполнение – это то, что можно делать с существующим файлом, возможные действия над ним. У каждой категории пользователей (владельца, группы, остальных) должны быть свои права на каждое вышеупомянутое действие.

- Право на чтение (r – read) означает, что файл можно просматривать. Например, открыть файл и, если он текстовый, прочитать содержащийся в нем текст. Если это файл изображения, то можно посмотреть изображение. Наличие права только на чтение не позволяет изменять файл. То есть нельзя будет исправить текст или подрисовать что-то к картинке.

- Право на запись (w – write) позволяет изменять файл, то есть дописывать в него информацию или заменять ее другой.

- Право на исполнение (x – execution) имеет смысл не для всех файлов, хотя может быть установлено для любого. Это право позволяет исполнять файл как программу, при этом в файле должны быть записаны инструкции для процессора, то есть файл должен быть исполняемой программой.

Первые три записи – это права владельца, вторые три записи – права группы, последняя тройка – права на файл для всех остальных. Если обозначить каждое право соответствующей буквой, и все права всем предоставляются, то получится такая запись: rwxrwxrwx [2].

Рассмотрим подробнее, что значат условные значения флагов прав:

— - нет прав, совсем;

–x - разрешено только выполнение файла, как программы но не изменение и не чтение;

-w- - разрешена только запись и изменение файла;

-wx - разрешено изменение и выполнение, но в случае с каталогом, вы не можете посмотреть его содержимое;

r– - права только на чтение;

r-x - только чтение и выполнение, без права на запись;

rw- - права на чтение и запись, но без выполнения;

rwx - все права;

–s - установлен SUID или SGID бит, первый отображается в поле для владельца, второй для группы;

–t - установлен sticky-bit, а значит пользователи не могут удалить этот файл [3].

Использование команды ls с опцией -l выведет на экран «длинную» распечатку, в которой будут, среди прочего, отражены права доступа к файлу [4].

Все группы, созданные в системе, находятся в файле /etc/group. Посмотрев содержимое этого файла, вы можете узнать список групп linux, которые уже есть в вашей системе.

Кроме стандартных root и users, здесь есть еще пару десятков групп. Это группы, созданные программами, для управления доступом этих программ к общим ресурсам. Каждая группа разрешает чтение или запись определенного файла или каталога системы, тем самым регулируя полномочия пользователя, а следовательно, и процесса, запущенного от этого пользователя. Здесь можно считать, что пользователь - это одно и то же что процесс, потому что у процесса все полномочия пользователя, от которого он запущен [5].

Расширенные атрибуты файловых объектов (далее - расширенные атрибуты) - поддерживаемая некоторыми файловыми системами возможность ассоциировать с файловыми объектами произвольные метаданные. В отличие от обычных атрибутов файловых объектов (таких, как владелец, права доступа, время создания и пр.), содержание расширенных атрибутов не специфицируется в файловой системе и может принимать любые значение. С точки зрения реализации расширенные атрибуты представляют собой пары ключ:значение, ассоциированные с файловыми объектами. Типичными применениями расширенных атрибутов является хранение таких данных, как автор документа, контрольные суммы, источник документа, информация для контроля доступа [6].

Есть три бита – Setuid, Setgid и Sticky Bit. Это специальные типы разрешений позволяют задавать расширенные права доступа на файлы или каталоги.

Setuid – это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо. Классический пример использования этого бита в операционной системе это команда sudo [7].

Принцип работы Setgid очень похож на setuid с отличием, что файл будет запускаться пользователем от имени группы, которая владеет файлом [7].

Последний специальный бит разрешения – это Sticky Bit . В случае, если этот бит установлен для папки, то файлы в этой папке могут быть удалены только их владельцем. Пример использования этого бита в операционной системе это системная папка /tmp . Эта папка разрешена на запись любому пользователю, но удалять файлы в ней могут только пользователи, являющиеся владельцами этих файлов [7].

4 Выполнение лабораторной работы

1. Создал программу simpleid.c и simpleid2.c (рис. 4.1).

```c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
        uid_t real_uid = getuid ();
        uid_t e_uid = geteuid ();

        gid_t real_gid = getgid ();
        gid_t e_gid = getegid ();

        printf("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
        printf("real_uid=%d, real_gid=%d\n", real_uid, real_gid);

        return 0;
}
```

Рис. 4.1: Код программы simpleid2.c

2. От имени суперпользователя выполнил команды: chown root:guest /home/guest/simpleid2 chmod u+s /home/guest/simpleid2. Запустил simpleid2 и id и сравнил результаты (рис. 4.2).

```
[guest@gabystrov ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@gabystrov ~]$ gcc simpleid2.c -o simpleid2
[guest@gabystrov ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@gabystrov ~]$
```

Рис. 4.2: Команды ./simpleid2 и id

3. Создайл программу readfile.c (рис. 4.3).

```c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

int
main (int argc, char* argv[])
{
        unsigned char buffer[16];
        size_t bytes_read;
        int i;

        int fd = open (argv[1], O_RDONLY);
        do
        {
                bytes_read = read (fd, buffer, sizeof (buffer));
                for (i = 0; i < bytes_read; ++i) printf("%c", buffer[i]);
        }

        while (bytes_read == sizeof (buffer));
        close (fd);
        return 0;
}
```

Рис. 4.3: Код программы readfile.c

4. Сменил владельца у файла readfile.c (или любого другого текстового файла в системе) и изменил права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (рис. 4.4).

```
[root@gabystrov guest]# chown root:guest /home/guest/readfile.c
[root@gabystrov guest]# chmod 700 /home/guest/readfile.c
[root@gabystrov guest]# su - guest
[guest@gabystrov ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@gabystrov ~]$ su
Пароль:
[root@gabystrov guest]# chown root:guest /home/guest/readfile
[root@gabystrov guest]# chmod u+s /home/guest/readfile
[root@gabystrov guest]# su - guest
```

Рис. 4.4: Работа с консолью

5. От имени пользователя guest создал файл file01.txt в директории /tmp со словом test (рис. 4.5).

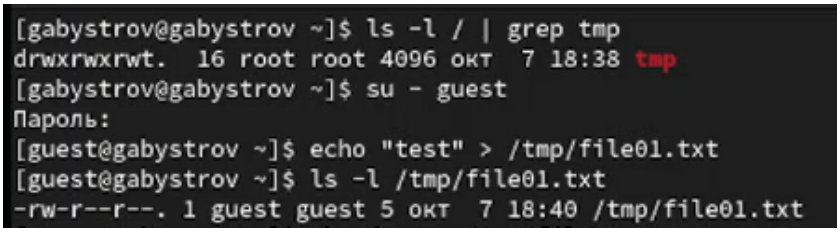

Рис. 4.5: Создание файла

6. Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей «все остальные» (рис. 4.6).

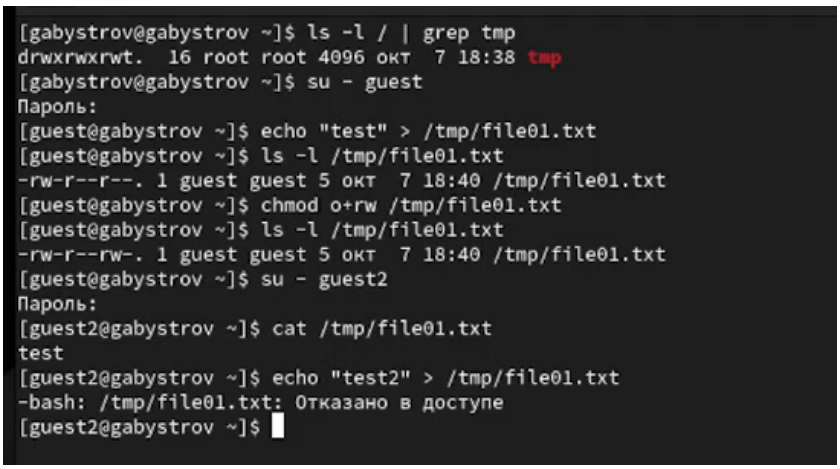

Рис. 4.6: Просмотр атрибутов

7. От пользователя guest2 попробовал записать в файл /tmp/file01.txt слова test2 и test3, стерев при этом всю имеющуюся в файле информацию (рис. 4.7).

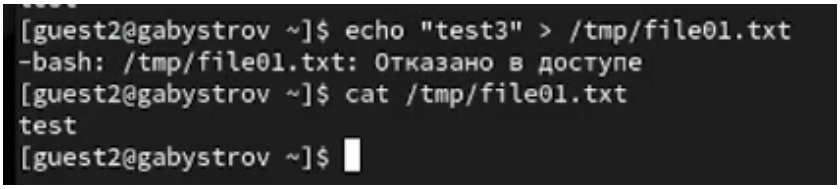

Рис. 4.7: Команда echo “test3” > /tmp/file01.txt

8. Повысил свои права до суперпользователя следующей командой su - и выполнил после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp (рис. 4.8).

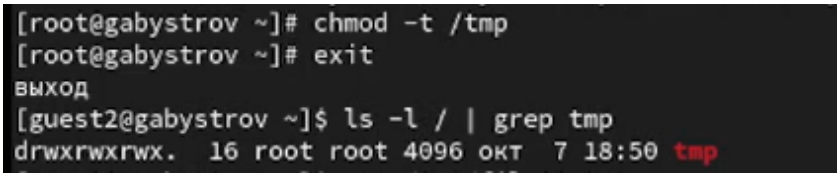

Рис. 4.8: Снятие атрибута

9. От пользователя guest2 проверил, что атрибута t у директории /tmp нет: ls -l / | grep tmp. Повторил предыдущие шаги. (рис. 4.9).

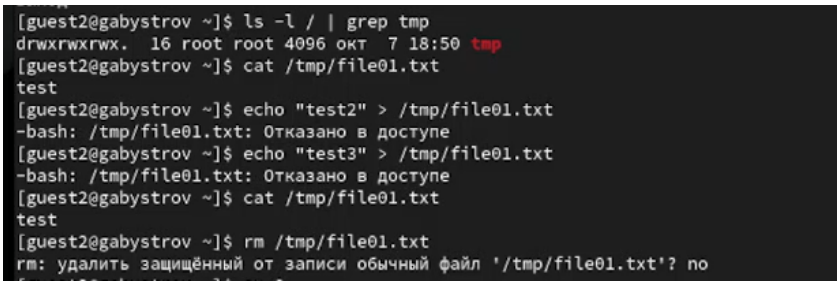

Рис. 4.9: Повторение шагов

10. Повысил свои права до суперпользователя и верните атрибут t на директорию /tmp (рис. 4.10).

Рис. 4.10: Добавление атрибута

5 Выводы

В данной лабораторной работе мне успешно удалось изучить механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получить практические навыки работы в консоли с дополнительными атрибутами. Рассмотреть работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Список литературы

1. Права в Linux (chown, chmod, SUID, GUID, sticky bit, ACL, umask) [Электронный ресурс]. 2023. URL: https://habr.com/ru/articles/469667/.
2. Права доступа к файлам и каталогам [Электронный ресурс]. 2023. URL: https://younglinux.info/bash/rwx.
3. Права доступа к файлам в Linux [Электронный ресурс]. 2023. URL: https://losst.pro/prava-dostupa-k-fajlam-v-linux.
4. Права доступа к файлам [Электронный ресурс]. 2023. URL: https://docs.altlinux.org/ru-RU/archive/2.3/html-single/junior/alt-docs-extras-linuxnovice/ch02s08.html.
5. Группы пользователей Linux [Электронный ресурс]. 2023. URL: https://losst.pro/gruppy-polzovatelej-linux.
6. Работа с расширенными атрибутами [Электронный ресурс]. 2023. URL: https://wiki.astralinux.ru/pages/viewpage.action?pageId=149063848.
7. Использование SETUID, SETGID и Sticky bit для расширенной настройки прав доступа в операционных системах Linux [Электронный ресурс]. 2023. URL: https://ruvds.com/ru/helpcenter/suid-sgid-sticky-bit-linux/.