Отчет по лабораторной работе №5

Информационная безопасность

Чекалова Лилия Руслановна

Содержание

Список таблиц

Список иллюстраций

1 Цель работы

- Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов
- Получение практических навыков работы в консоли с дополнительными атрибутами
- Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

2 Задание

- Написание программ
- Изменение владельца файлов и прав доступа на файлы
- Установка и снятие Sticky-бита и проверка доступных действий

3 Теоретическое введение

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.

На месте, где обычно установлен классический бит x (на исполнение), выставлен специальный бит s. Это позволяет обычному пользователю системы выполнять команды с повышенными привилегиями без необходимости входа в систему как root, зная пароль пользователя root.

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

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.

Более подробно о см. в [1,2].

4 Выполнение лабораторной работы

В качестве первого шага лабораторной работы мы осуществили вход от лица пользователя guest и создали файл simpleid.c (рис. 4.1).

```
[lrchekalova@lrchekalova guest]$ su guest
Пароль:
[guest@lrchekalova ~]$ touch simpleid.c
[guest@lrchekalova ~]$
```

Рис. 4.1: Создание файла

Далее мы написали программу, которая выводитинформацию об идентификаторах пользователя и группы (рис. 4.2). Скомпилировали эту программу (рис. 4.3).

```
1  #include <sys/types.h>
2  #include <unistd.h>
3  #include <stdio.h>
4
5  int
6  main()
7  {
8      uid_t uid = geteuid();
9      gid_t gid = getegid();
10     printf("uid=%d, gid=%d\n", uid, gid);
11     return 0;
12 }
```

Рис. 4.2: Программа 1

```
[guest@lrchekalova ~]$ gcc simpleid.c -o simpleid
[guest@lrchekalova ~]$ ls
dir1  simpleid  simpleid.c
[guest@lrchekalova ~]$
```

Рис. 4.3: Компиляция программы 1

Выполнили эту программу и сравнили ее вывод с результатом работы системной команды id (рис. 4.4). Результаты сходятся, но команда id дает дополнительные сведения, например, название группы.

```
[guest@lrchekalova ~]$ ./simpleid
uid=1001, gid=1001
[guest@lrchekalova ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@lrchekalova ~]$
```

Рис. 4.4: Запуск программы 1

Модифицировали программу, чтобы она выводила также действительные идентификаторы (рис. 4.5).

```
#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. 4.5: Программа 2

Скомпилировали и запустили модифицированную программу (рис. 4.6).

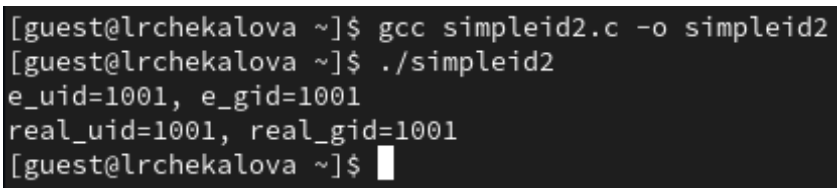

Рис. 4.6: Запуск модифицированной программы

Передали право владения программой суперпользователю и наделили его правом на исполнение программы. Запустили программу и сверили результат с выводом команды id (рис. 4.7). Теперь в качестве идентификаторов указана информация root-пользователя.

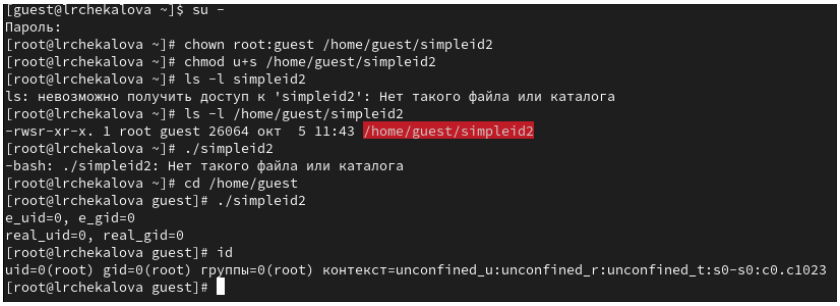

Рис. 4.7: Передача прав и перезапуск программы

Проделали то же самое относительно SetGID-бита (рис. 4.8).

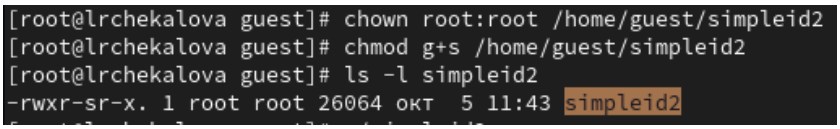

Рис. 4.8: Повтор действий относительно SetGID-бита

Создали новую программу readfile, позволяющую прочесть содержимое файла (рис. 4.9).

```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. 4.9: Программа 3

Скомпилировали ее, а затем поменяли владельца файла readfile.c (рис. 4.10) и изменили права так, чтобы читать файл мог только суперпользователь. Проверили, может ли пользователь guest прочитать файл (рис. 4.11). Ему было отказано в доступе.

```
[root@lrchekalova guest]# chown root:root readfile.c
[root@lrchekalova guest]# chmod u+s readfile.c
[root@lrchekalova guest]# ls -l readfile.c
-rwSr--r--. 1 root root 456 окт  5 12:05 readfile.c
```

Рис. 4.10: Смена владельца файла

```
[root@lrchekalova ~]# chmod o-r /home/guest/readfile.c
[root@lrchekalova ~]# exit
выход
[guest@lrchekalova ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@lrchekalova ~]$
```

Рис. 4.11: Смена прав и проверка

Передали право владения программой суперпользователю и проверили, может ли программа прочитать readfile.c (рис. 4.12) и etc/shadow (рис. 4.13). Действия были выполнены успешно.

```
[guest@lrchekalova ~]$ su -
Пароль:
[root@lrchekalova ~]# chown root:root /home/guest/readfile
[root@lrchekalova ~]# chmod u+s /home/guest/readfile
[root@lrchekalova ~]# ./readfile readfile.c
-bash: ./readfile: Нет такого файла или каталога
[root@lrchekalova ~]# cd /home/guest
[root@lrchekalova guest]# ./readfile readfile.c
#include <fcntl.h>
#include <stdio.h>
```

Рис. 4.12: Смена владельца программой и чтение readfile.c

```
[root@lrchekalova guest]# ./readfile /etc/shadow
root:$6$HiPuTZPpUOs5jFtY$xjdDaoqwUV7zpkuYu/Vqbtjq3y
bin:*:19469:0:99999:7:::
daemon:*:19469:0:99999:7:::
```

Рис. 4.13: Чтение etc/shadow

Проверили, установлен ли Sticky-бит на директории tmp, создали file01.txt и передали категории пользователей “все остальные” право на чтение и запись (рис. 4.14).

```
[guest@lrchekalova ~]$ ls -l / | grep tmp
drwxrwxrwt.  17 root root 4096 окт  5 12:29 tmp
[guest@lrchekalova ~]$ echo "test" > /tmp/file01.txt
[guest@lrchekalova ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 окт  5 12:30 /tmp/file01.txt
[guest@lrchekalova ~]$ chmod o+rw /tmp/file01.txt
[guest@lrchekalova ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 окт  5 12:30 /tmp/file01.txt
[guest@lrchekalova ~]$
```

Рис. 4.14: Проверка наличия Sticky-бита и создание файла

От лица пользователя guest2 (не являющегося владельцем файла) попробовали прочесть файл и записать туда новые данные (рис. 4.15), а также удалить файл (рис. 4.16). Мы

смогли прочесть файл, но не смогли изменить его и удалить, так как guest2 входит в группу guest, а не в группу “все остальные”.

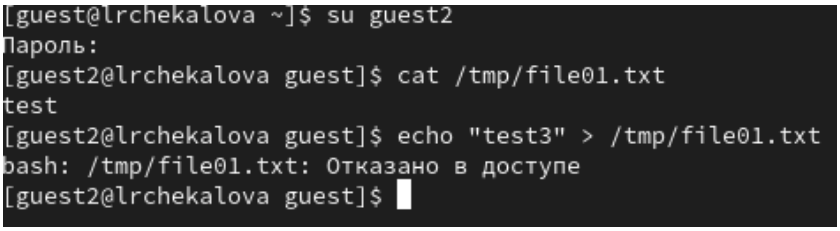

Рис. 4.15: Попытка чтения и изменения

Рис. 4.16: Попытка удаления

Сняли с папки tmp атрибут Sticky (рис. 4.17).

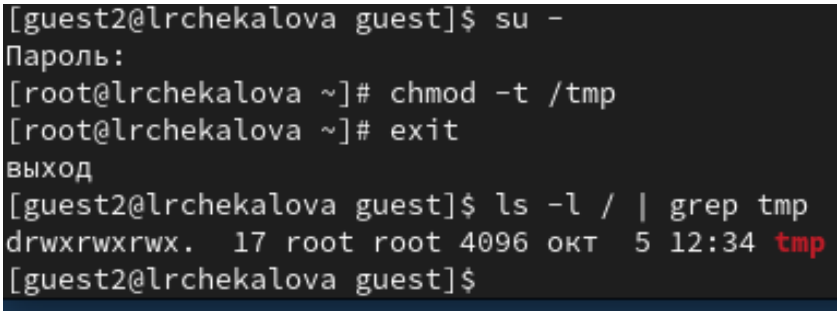

Рис. 4.17: Снятие атрибута Sticky

От лица пользователя guest2 попробовали прочесть файл, произвести в него запись и удалить (рис. 4.18). Удаление файла прошло успешно, так как с папки был снят атрибут, запрещавший удаление файлов всеми, кроме владельца.

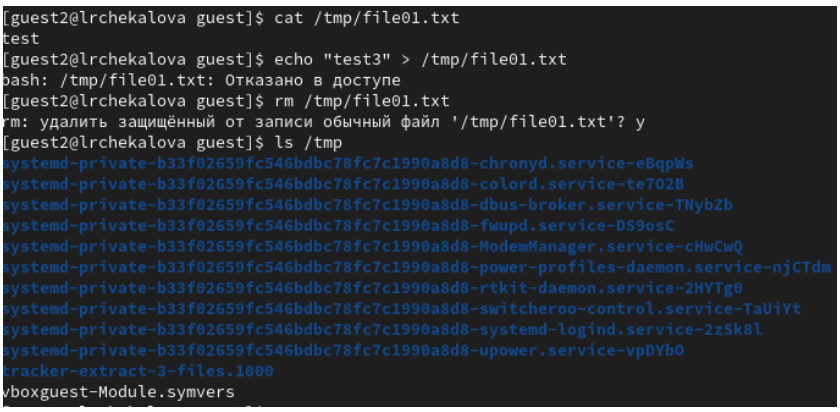

Рис. 4.18: Чтение, запись и удаление

Установили атрибут Sticky обратно на папку tmp (рис. 4.19).

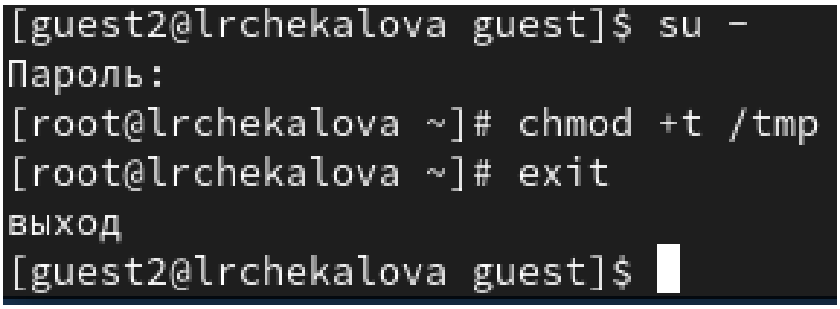

Рис. 4.19: Установка атрибута Sticky

5 Выводы

В результате лабораторной работы я повысила навыки использования командой строки и изучила SetUID-, SetGID- и Sticky-биты. Также я рассмотрела работу механизма смены идентификатора процессов пользователей.

Список литературы

1. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов [Электронный ресурс]. URL: https://esystem.rudn.ru/mod/resource/view.php?id=1031377.
2. Использование SETUID, SETGID и Sticky Bit для расширенной настройки прав доступа в операционных системах Linux [Электронный ресурс]. URL: https://ruvds.com/ru/helpcenter/suid-sgid-sticky-bit-linux/.