Отчет по лабораторной работе №5

Информационная безопасность

Байрамгельдыев Довлетмурат

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

- Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов
- Получение практических навыков работы в консоли с дополнительными атрибутами

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 [**lab-theory?**,**ruvds?**].

4 Выполнение лабораторной работы

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

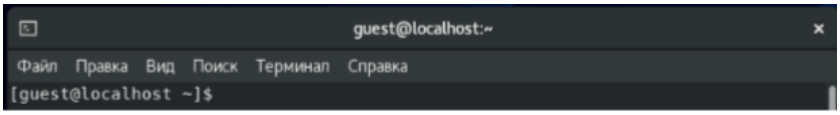


Рис. 4.1: Создание файла

Далее написал программу, которая выводит информацию об идентификаторах пользователя и группы (рис. 4.2). Скомпилировал эту программу (рис. 4.3).

simpleid.c

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

int
main ()
{
  uid_t uid = geteuid ();
  gid_t gid = getegid ();
  printf ("uid=%d, gid=%d\n", uid, gid);
  return 0;
}
```

Рис. 4.2: Программа 1

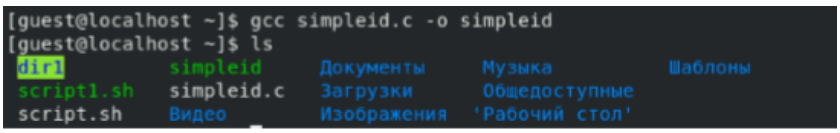
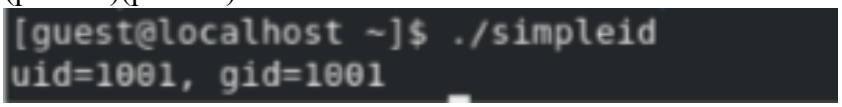


Рис. 4.3: Компиляция программы 1

Выполнил эту программу и сравнил ее вывод с результатом работы системной команды id (рис. **??**)(рис. **??**).



Модифицировал программу, чтобы она выводила также действительные идентификаторы (рис. 4.4).

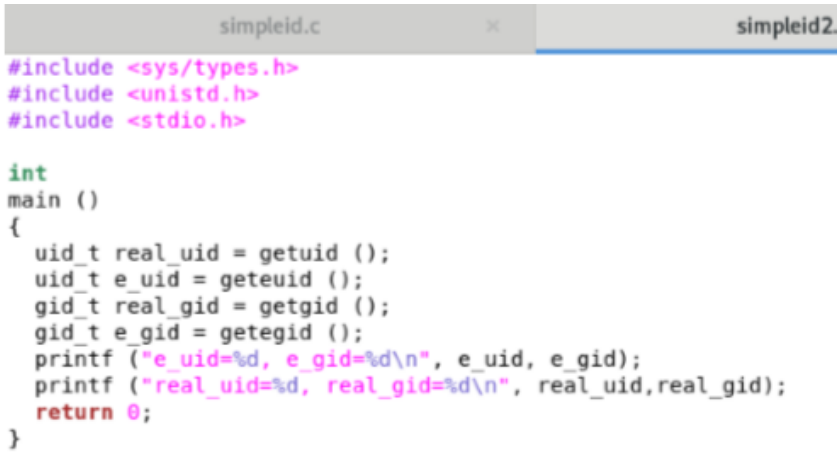


Рис. 4.4: Программа 2

Скомпилировал и запустил модифицированную программу (рис. 4.5).

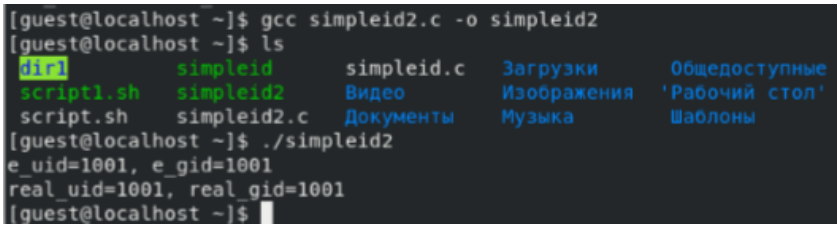
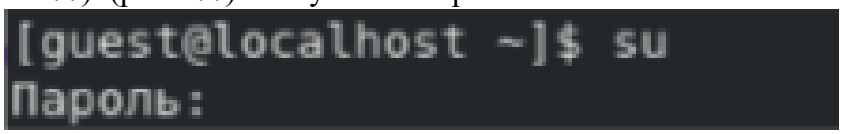


Рис. 4.5: Запуск модифицированной программы

Передал право владения программой суперпользователю и наделил его правом на исполнение программы. Запустили программу и сверили результат с выводом команды id (рис. **??**) (рис. **??**). Запустил simpleid2 и id. Ничего не изменилось (рис. **??**)



Создал новую программу readfile, позволяющую прочесть содержимое файла (рис. 4.6).

simpleid.c | simpleid2.c | readfile.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. 4.6: Программа 3

Скомпилировал ее, а затем поменял владельца файла readfile.c (рис. **??**) и изменил права так, чтобы читать файл мог только суперпользователь. Проверил, может ли пользователь guest прочитать файл (рис. **??**). Ему было отказано в доступе.

Передал право владения программой суперпользователю и проверили, может ли программа прочитать readfile.c (рис. **??**) (рис. **??**) и etc/shadow (рис. **??**). Действия были

выполнены успешно.

```
[guest@localhost ~]$ ./reac
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
```

```
[root@localhost guest]# chown root:guest /home/guest/readfile
[root@localhost guest]# chmod u+s /home/guest/readfile
```

```
[guest@localhost ~]$ ./readfile /etc/shadow
root:$6$0C3WkySzs7.eemZD$bktpYmRhX3z7SHEO4ngRFdRsDpJUgLfiLs4P8Pj3TvWWATuRTWBuX38
9ZOUw25fVDJu7zzupnftxRrco5QO9j.::0:99999:7:::
bin:*:18358:0:99999:7:::
daemon:*:18358:0:99999:7:::
adm:*:18358:0:99999:7:::
lp:*:18358:0:99999:7:::
sync:*:18358:0:99999:7:::
shutdown:*:18358:0:99999:7:::
halt:*:18358:0:99999:7:::
mail:*:18358:0:99999:7:::
operator:*:18358:0:99999:7:::
games:*:18358:0:99999:7:::
```

Проверил, установлен ли Sticky-бит на директории tmp, создал file01.txt и передал категории пользователей “все остальные” право на чтение и запись (рис. **??**) (рис. **??**).

```
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$ cat /tmp/file01.txt
test
```

```
[guest2@localhost guest]$ cat /tmp/file01.txt
test
```

От лица пользователя guest2 (не являющегося владельцем файла) попробовал прочесть файл и записать туда новые данные (рис. **??**) (рис. **??**), а также удалить файл (рис. **??**).

```
[guest2@localhost guest]$ cat /tmp/file01.txt
test
test2
```

```
[guest2@localhost guest]$ echo "test
[guest2@localhost guest]$
```

```
[guest2@localhost guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
```

Снял с папки tmp атрибут Sticky (рис. **??**) (рис. **??**).

```
[guest2@localhost guest]$ su
Пароль:
[root@localhost guest]# chmod -t /tmp
```

```
[root@localhost guest
exit
```

От лица пользователя guest2 попробовал прочесть файл, произвести в него запись и удалить (рис. 4.7). Удаление файла прошло успешно.

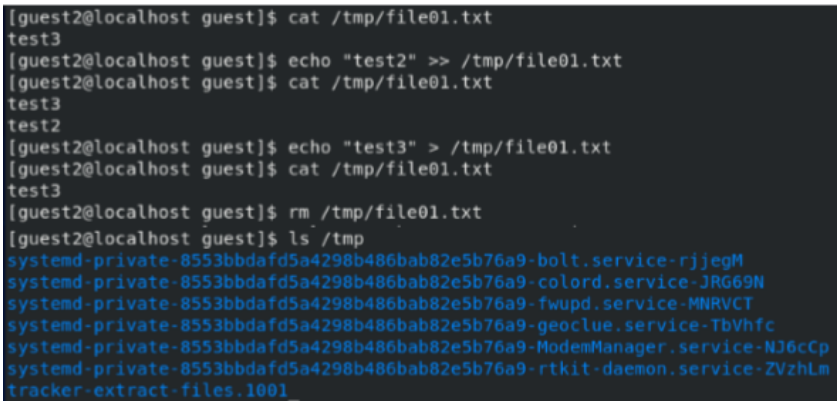
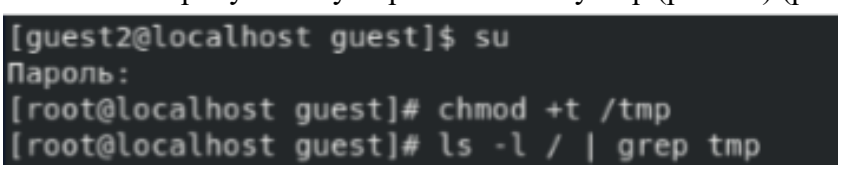
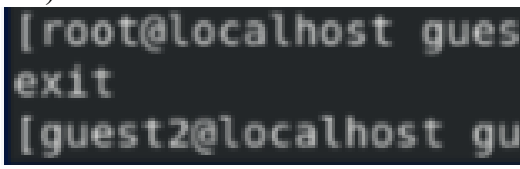


Рис. 4.7: Чтение, запись и удаление

Установил атрибут Sticky обратно на папку tmp (рис. **??**) (рис. **??**).





5 Выводы

В результате лабораторной работы усовершенствовал навыки использования командой строки и изучил SetUID-, SetGID- и Sticky-биты.

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/.