

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Назарьева Алена НФИбд-03-18

2021, 9 november

inst{1}RUDN University, Moscow, Russian Federation


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

Выполнение лабораторной работы

1)

Вошла в систему от имени пользователя guest. Создала программу simpleid.c (рис. -fig. 1)



```
#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;
}
```


Figure 1: simpleid.c

Скомпилировала программу и убедилась, что файл программы создан. Выполнила программу simpleid. Выполнила системную программу id и сравнила полученный результат с данными предыдущего пункта задания. Данные совпадают (рис. -fig. 2)

```
[guest@ainazarieva ~]$ gcc simpleid.c -o simpleid
[guest@ainazarieva ~]$ ./simpleid
uid=1001, gid=1001
[guest@ainazarieva ~]$ id
uid=1001(guest) gid=1001(guest) rpyнны=1001(guest) контекст=unconfined_u:unconfined_r:u
nconfined_t:s0-s0:c0.c1023
[guest@ainazarieva ~]$
```

Figure 2: пункты 3-5

Усложнила программу, добавив вывод действительных идентификаторов. Получившуюся программу назвала simpleid2.c. (рис. -fig. 3)



```
#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;
}
```

Figure 3: simpleid2.c

Скомпилировала и запустила simpleid2.c. От имени суперпользователя сменила у файла владельца и установила установите SetU'D-бит. Использовала sudo или повысила временно свои права с помощью su.(рис. -fig. 4)

```
[guest@ainazarieva ~]$ gcc simpleid2.c -o simpleid2
[guest@ainazarieva ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@ainazarieva ~]$ su
Пароль:
[root@ainazarieva guest]# chown root:guest /home/guest/simpleid2
[root@ainazarieva guest]# chmod u+s /home/guest/simpleid2
[root@ainazarieva guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8576 ноя  5 11:06 simpleid2
[root@ainazarieva guest]#
```

Figure 4: пункты 7-9

Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2. Запустила simpleid2 и id. Результаты совпадают. (рис. -fig. 5)

```
[guest@ainazarieva ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 8576 ноя  5 11:06 simpleid2
[guest@ainazarieva ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@ainazarieva ~]$ id
uid=1001(guest) gid=1001(guest) rpyнны=1001(guest) контекст=unconfined_u:unconfined_r:u
nconfined_t:s0-s0:c0.c1023
[guest@ainazarieva ~]$
```

Figure 5: пункты 10-11

Проделала тоже самое относительно SetGID-бита. (рис. -fig. 6)

```
[guest@ainazarieva ~]$ su
Пароль:
[root@ainazarieva guest]# chown root:guest /home/guest/simpleid2
[root@ainazarieva guest]# chmod g+s /home/guest/simpleid2
[root@ainazarieva guest]# ls -l simpleid2
-rwxrwsr-x. 1 root guest 8576 ноя  5 11:06 simpleid2
```

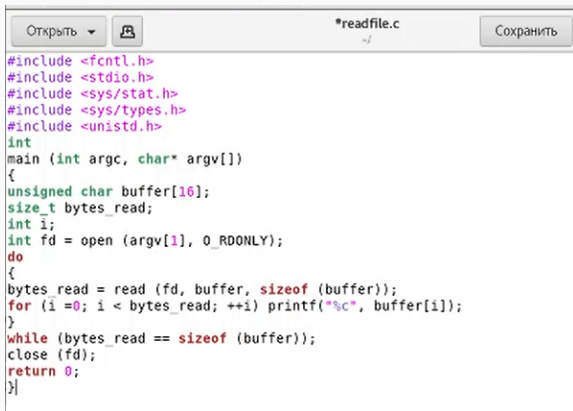
Figure 6: пункты 12-1

(рис. -fig. 7)

```
[guest@ainazarieva ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@ainazarieva ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:u
nconfined_t:s0-s0:c0.c1023
[guest@ainazarieva ~]$
```

Figure 7: пункты 12-2

Создала программу readfile.c (рис. -fig. 8)



```
#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;
}
```

Figure 8: readfile.c

Откомпилировала её. Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь мог прочитать его, а guest не мог. (рис. -fig. 9)

```
пароль:  
[root@ainazarieva guest]# touch readfile1.c  
[root@ainazarieva guest]# chmod 700 /home/guest/readfile1.c  
[root@ainazarieva guest]# █
```

Figure 9: пункты 14-15

Проверила, что пользователь guest не может прочитать файл readfile.c. (рис. -fig. 10)

```
[guest@ainazarieva ~]$ cat readfile.c  
cat: readfile.c: Отказано в доступе  
[guest@ainazarieva ~]$ █
```

Figure 10: пункт 16

Сменила у программы readfile владельца и установила SetU'D-бит. Проверила, может ли программа readfile прочитать файл readfile.c (рис. -fig. 11)

```
[root@ainazarieva guest]# chown root:root /home/guest/readfile
[root@ainazarieva guest]# chmod u+s /home/guest/readfile
[root@ainazarieva guest]# ./readfile
Содержимое файла:rn#include <stdio.h>
#include <fcntl.h>
#include <string.h>

int main(int argc, char *argv[])
{
    char buff[2048];
    //Открытие файла
    int file = open("readfile.c", O_RDWR);
    //Вывод содержимого
    read(file, buff, 2048);
    printf("Содержимое файла:rn%srn", buff);
    //Обнуляем буфер
    memset(buff, 0, 2048);
    close(file);
}

rn[root@ainazarieva guest]# █
```

Figure 11: пункты 17-18

Проверила, может ли программа readfile прочитать файл /etc/shadow (рис. -fig. 12)

```
[root@ainazarieva guest]# ./readfile
Содержимое файла:rnroot:$6$BLHeo/qhivR5F2Jp$guE5DxIFeA5QTloLyF4fzZ6RNDf0DPLlGPNYjE7BMxd
rbW436610Ht680llsCLF15akZ5CDh7xo2m8dzEgL9/::0:99999:7:::
bin:*:18353:0:99999:7:::
daemon:*:18353:0:99999:7:::
adm:*:18353:0:99999:7:::
lp:*:18353:0:99999:7:::
sync:*:18353:0:99999:7:::
shutdown:*:18353:0:99999:7:::
halt:*:18353:0:99999:7:::
mail:*:18353:0:99999:7:::
```

Figure 12: пункт 19

Так как владелец файла `readfile` `root`, а `setuid` являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами владельца исполняемого файла, то `readfile` смог прочитать и `readfile.c` и `/etc/shadow`

Исследование Sticky-бита. Выяснила, что установлен атрибут Sticky на директории /tmp. От имени пользователя guest создала файл file01.txt в директории /tmp со словом test. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные» (рис. -fig. 13)

```
[guest@ainazarieva ~]$ echo "test" > /tmp/file01.txt
[guest@ainazarieva ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя  8 18:30 /tmp/file01.txt
[guest@ainazarieva ~]$ chmod o+rw /tmp/file01.txt
[guest@ainazarieva ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя  8 18:30 /tmp/file01.txt
[guest@ainazarieva ~]$ █
```

Figure 13: пункт 1-3

От пользователя guest2 (не являющегося владельцем) попробовала прочитать файл /tmp/file01.txt От пользователя guest2 не смогла дозаписать в файл. Проверила содержимое файла. От пользователя guest2 записала в файл /tmp/file01.txt слово test3, стеревав при этом всю имеющуюся в файле информацию. Проверила содержимое файла. От пользователя guest2 не смогла удалить файл /tmp/file01.txt. Повысила свои права до суперпользователя следующей и выполнила после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp. Покинула режим суперпользователя

От пользователя guest2 проверила, что атрибута t у директории /tmp нет (рис. -fig. 14)

```
[guest2@ainazarieva ~]$ cat /tmp/file01.txt
test
[guest2@ainazarieva ~]$ echo "test" > /tmp/file01.txt
[guest2@ainazarieva ~]$ echo "test2" > /tmp/file01.txt
[guest2@ainazarieva ~]$ cat /tmp/file01.txt
test2
[guest2@ainazarieva ~]$ echo "test3" > /tmp/file01.txt
[guest2@ainazarieva ~]$ cat /tmp/file01.txt
test3
[guest2@ainazarieva ~]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
[guest2@ainazarieva ~]$ su -
Пароль:
Последний вход в систему: Пт ноя  5 12:34:31 MSK 2021 на pts/0
[root@ainazarieva ~]# chmod -t /tmp
[root@ainazarieva ~]# exit
logout
[guest2@ainazarieva ~]$ ls -l / | grep tmp
drwxrwxrwx. 34 root root 4096 ноя  8 18:37 tmp
```

Figure 14: пункт 4-12

Повторила предыдущие шаги. Мне удалось удалить файл от имени пользователя, не являющегося его владельцем. Благодаря Sticky bit пользователи могут создавать файлы, читать и выполнять их, принадлежащие другим пользователям, но не могут удалять файлы, принадлежащие другим пользователям, даже если в каталоге есть разрешение 777. Если sticky bit не установлен, то юзер может удалить файл, так как он наследует разрешения родительского каталога. Повысила свои права до суперпользователя и вернула атрибут t на директорию /tmp (рис. -fig. 15)

```
[guest2@ainazarieva ~]$ cat /tmp/file01.txt
test3
[guest2@ainazarieva ~]$ echo "test2" > /tmp/file01.txt
[guest2@ainazarieva ~]$ cat /tmp/file01.txt
test2
[guest2@ainazarieva ~]$ rm /tmp/file01.txt
[guest2@ainazarieva ~]$ su -
Пароль:
```

Выводы

В результате выполнения работы я Изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов