

Лабораторная работа № 5

Юрченко Артём Алексеевич

2023, Москва

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.

1. Исследовать SetUID- и SetGID-биты.
2. Исследовать Sticky-бит.

От имени пользователя guest создадим программу simpleid.c, скомпилируем ее и убедимся, что файл создан.

```
[guest@info-sec ~]$ touch simpleid.c  
[guest@info-sec ~]$ gcc simpleid.c -o simpleid
```

Рис. 1: Создание файла simpleid.c

Выполним команды `./simpleid` и `id` и убедимся, что полученные данные совпадают.

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

Рис. 2: Использование команд `./simpleid` и `id`

Усложним программу и запишем ее в файл `simpleid2.c`. Запустим получившуюся программу.

```
[guest@info-sec ~]$ touch simpleid2.c
[guest@info-sec ~]$ gcc simpleid2.c -o simpleid2
[guest@info-sec ~]$ ./simpleid2
e_uid=1001, e_gid=1001          I
real_uid=1001, real_gid=1001
```

Рис. 3: Создание и запуск программы `simpleid2`

От имени суперпользователя установим новые атрибуты и сменим владельца файла simpleid2.

```
[guest@info-sec ~]$ su
Пароль:
[root@info-sec guest]# chown root:guest /home/guest/simpleid2
[root@info-sec guest]# chmod u+s /home/guest/simpleid2
[root@info-sec guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26064 окт  7 19:36 simpleid2
```

Рис. 4: Установки новых атрибутов и смена владельца файла simpleid2

Выполним команды `./simpleid2` и `id` и убедимся, что полученные данные совпадают.

```
[root@info-sec guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@info-sec guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@info-sec guest]#
```

Рис. 5: Использование команд `./simpleid2` и `id`

Проделаем то же самое относительно SetGID-бита.

```
[root@user progs]# chmod g+s simpleid2
[root@user progs]# ls -l simpleid2
-rwsr-sr-x. 1 root guest 26064 Oct  5 14:23 simpleid2
[root@user progs]# exit
exit
[guest@user progs]$ ^M
: command not found...
[guest@user progs]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@user progs]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

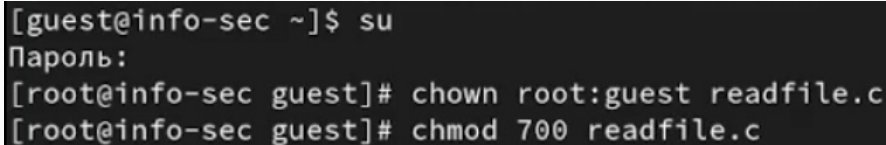
Рис. 6: Операции с SetGID-битом

Создадим и скомпилируем программу readfile.c.

```
[guest@info-sec ~]$ touch readfile.c  
[guest@info-sec ~]$ gcc readfile.c -o readfile
```

Рис. 7: Создание и компиляция программы readfile.c

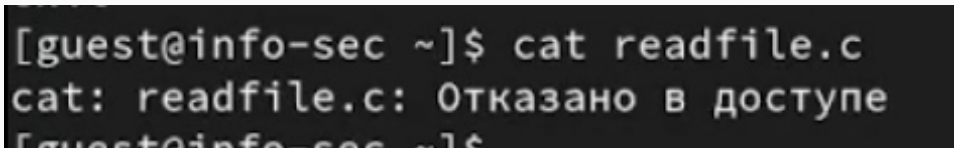
Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.



```
[guest@info-sec ~]$ su
Пароль:
[root@info-sec guest]# chown root:guest readfile.c
[root@info-sec guest]# chmod 700 readfile.c
```

Рис. 8: Изменение владельца и прав файла readfile.c

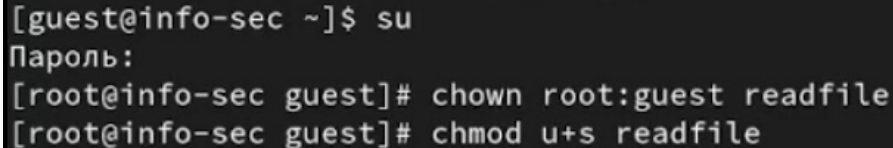
Проверим, что пользователь guest не может прочитать файл readfile.c.

A terminal window with a dark background and light gray text. The prompt is [guest@info-sec ~]\$. The user enters the command cat readfile.c. The output is cat: readfile.c: Отказано в доступе. The prompt is partially visible on the next line as [guest@info-sec ~]\$.

```
[guest@info-sec ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@info-sec ~]$
```

Рис. 9: Проверка, что пользователь guest не может прочитать файл readfile.c.

Сменим у программы readfile владельца и установим SetUID-бит.



```
[guest@info-sec ~]$ su
Пароль:
[root@info-sec guest]# chown root:guest readfile
[root@info-sec guest]# chmod u+s readfile
```

Рис. 10: Работа с параметрами readfile

Проверим, может ли программа readfile прочитать файл readfile.c.

```
[root@info-sec guest]# ./readfile 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];
```

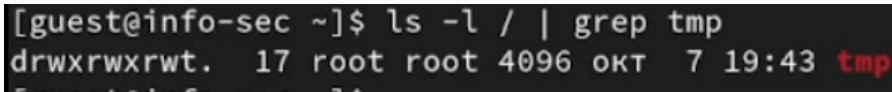
Рис. 11: Попытка прочитать файл readfile.c программой readfile

Проверим, может ли программа readfile прочитать файл /etc/shadow.

```
[root@info-sec guest]# ./readfile /etc/shadow
root:$6$irGYpLkCLN1K0rj0$g6wUExqhghqnlzil/Z5Ic0eXoGo5pDPTm34xd61HjWpqm0Vy9dyqD
N2Bf7Q2Dg/xC93iCM.G/ICpgUK/BVCUL1::0:99999:7:::
bin:*:19469:0:99999:7:::
daemon:*:19469:0:99999:7:::
adm:*:19469:0:99999:7:::
lp:*:19469:0:99999:7:::
sync:*:19469:0:99999:7:::
shutdown:*:19469:0:99999:7:::
halt:*:19469:0:99999:7:::
mail:*:19469:0:99999:7:::
operator:*:19469:0:99999:7:::
games:*:19469:0:99999:7:::
ftp:*:19469:0:99999:7:::
nobody:*:19469:0:99999:7:::
systemd-coredump:!!:19616:::
dbus:!!:19616:::
polkitd:!!:19616:::
```

Рис. 12: Попытка прочитать файл /etc/shadow программой readfile

Выясним, установлен ли атрибут Sticky на директории /tmp.



```
[guest@info-sec ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 окт 7 19:43 tmp
```

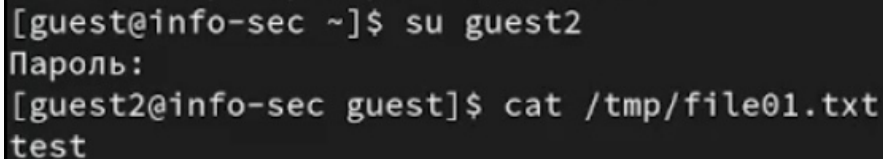
Рис. 13: Чтение атрибутов директории /tmp

От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные».

```
[guest@info-sec ~]$ echo "test" > /tmp/file01.txt
[guest@info-sec ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 окт  7 19:45 /tmp/file01.txt
[guest@info-sec ~]$ chmod o+rw /tmp/file01.txt
[guest@info-sec ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 окт  7 19:45 /tmp/file01.txt
```

Рис. 14: Чтение атрибутов директории /tmp

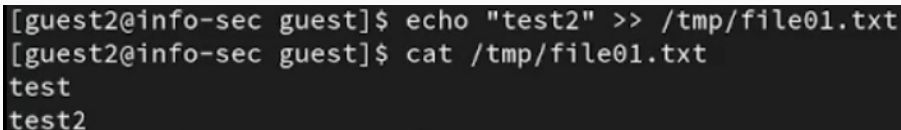
От пользователя guest2 попробуем прочитать файл /tmp/file01.txt.



```
[guest@info-sec ~]$ su guest2
Пароль:
[guest2@info-sec guest]$ cat /tmp/file01.txt
test
```

Рис. 15: Попытка прочтения файла /tmp/file01.txt

От пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2.



```
[guest2@info-sec guest]$ echo "test2" >> /tmp/file01.txt  
[guest2@info-sec guest]$ cat /tmp/file01.txt  
test  
test2
```

Рис. 16: Попытка дозаписи в файл /tmp/file01.txt

От пользователя guest2 попробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию).

```
[guest2@info-sec guest]$ echo "test3" > /tmp/file01.txt  
[guest2@info-sec guest]$ cat /tmp/file01.txt  
test3
```

Рис. 17: Попытка записи в файл /tmp/file01.txt

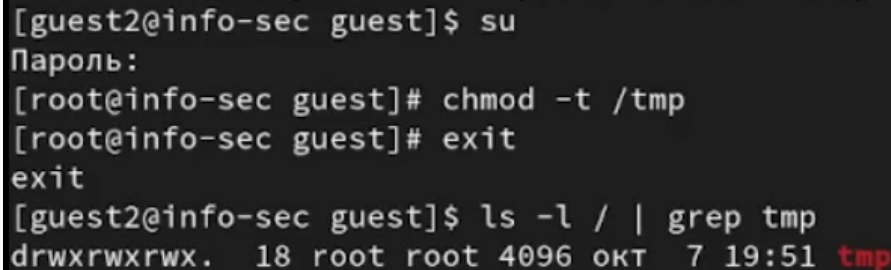
От пользователя guest2 попробуем удалить файл /tmp/file01.txt.

A terminal window with a dark background. The prompt is [guest2@info-sec guest]\$. The command entered is rm /tmp/file01.txt. The output is rm: невозможно удалить '/tmp/file01.txt': Операция не позволена.

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

Рис. 18: Попытка удаления файла /tmp/file01.txt

От имени суперпользователя снимем атрибут `t` с директории `/tmp`. От пользователя `guest2` проверим, что атрибута `t` у директории `/tmp` нет.



```
[guest2@info-sec guest]$ su
Пароль:
[root@info-sec guest]# chmod -t /tmp
[root@info-sec guest]# exit
exit
[guest2@info-sec guest]$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 окт  7 19:51 tmp
```

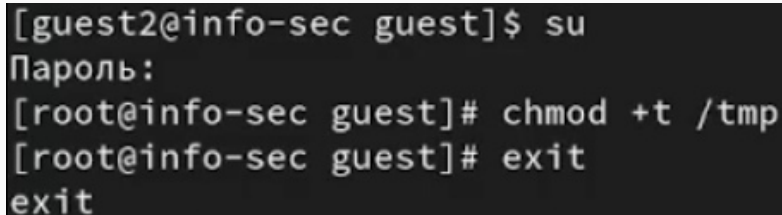
Рис. 19: Удаление атрибута `t` директории `/tmp`

Повторим предыдущие шаги. Теперь мы можем удалить файл.

```
[guest2@info-sec guest]$ cat /tmp/file01.txt
test3
[guest2@info-sec guest]$ echo "test2" >> /tmp/file01.txt
[guest2@info-sec guest]$ cat /tmp/file01.txt
test3
test2
[guest2@info-sec guest]$ echo "test3" > /tmp/file01.txt
[guest2@info-sec guest]$ cat /tmp/file01.txt
test3
[guest2@info-sec guest]$ rm /tmp/file01.txt
```

Рис. 20: Повторение предыдущих шагов

Повысим свои права до суперпользователя и вернем атрибут `t` на директорию `/tmp`.

A terminal window with a dark background and light-colored text. The text shows a user switching from 'guest' to 'root' using 'su', then using 'chmod +t /tmp' to set the sticky bit on the /tmp directory, and finally exiting the root shell with 'exit'.

```
[guest2@info-sec guest]$ su
Пароль:
[root@info-sec guest]# chmod +t /tmp
[root@info-sec guest]# exit
exit
```

Рис. 21: Возвращение атрибута `t` директории `/tmp`

В рамках данной лабораторной работы были изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами. Рассмотрены принципы работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.