

Лабораторная работа № 5

**Дискреционное разграничение прав в Linux. Исследование влияния
дополнительных атрибутов**

Юрченко Артём Алексеевич

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Выводы	15
	Список литературы	16

Список иллюстраций

4.1	Создание файла <code>simpleid.c</code>	7
4.2	Использование команд <code>./simpleid</code> и <code>id</code>	7
4.3	Создание и запуск программы <code>simpleid2</code>	8
4.4	Установки новых атрибутов и смена владельца файла <code>simpleid2</code> .	8
4.5	Использование команд <code>./simpleid2</code> и <code>id</code>	8
4.6	Операции с <code>SetGID</code> -битом	9
4.7	Создание и компиляция программы <code>readfile.c</code>	9
4.8	Изменение владельца и прав файла <code>readfile.c</code>	9
4.9	Проверка, что пользователь <code>guest</code> не может прочитать файл <code>readfile.c</code> .	9
4.10	Работа с параметрами <code>readfile</code>	10
4.11	Попытка прочитать файл <code>readfile.c</code> программой <code>readfile</code>	10
4.12	Попытка прочитать файл <code>/etc/shadow</code> программой <code>readfile</code>	11
4.13	Чтение атрибутов директории <code>/tmp</code>	11
4.14	Чтение атрибутов директории <code>/tmp</code>	11
4.15	Попытка прочтения файла <code>/tmp/file01.txt</code>	12
4.16	Попытка дозаписи в файл <code>/tmp/file01.txt</code>	12
4.17	Попытка записи в файл <code>/tmp/file01.txt</code>	12
4.18	Попытка удаления файла <code>/tmp/file01.txt</code>	12
4.19	Удаление атрибута <code>t</code> директории <code>/tmp</code>	13
4.20	Повторение предыдущих шагов	13
4.21	Возвращение атрибута <code>t</code> директории <code>/tmp</code>	14

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

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

3 Теоретическое введение

- Операционная система — это комплекс программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем [1].
- Права доступа определяют, какие действия конкретный пользователь может или не может совершать с определенным файлами и каталогами. С помощью разрешений можно создать надежную среду — такую, в которой никто не может поменять содержимое ваших документов или повредить системные файлы. [2].

4 Выполнение лабораторной работы

1. От имени пользователя guest создадим программу simpleid.c, скомпилируем ее и убедимся, что файл создан (4.1).

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

Рис. 4.1: Создание файла simpleid.c

2. Выполним команды ./simpleid и id и убедимся, что полученные данные совпадают (4.2).

```
[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. 4.2: Использование команд ./simpleid и id

3. Усложним программу и запишем ее в файл simpleid2.c. Запустим получившуюся программу (4.3).

```
[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. 4.3: Создание и запуск программы simpleid2

4. От имени суперпользователя установим новые атрибуты и сменим владельца файла simpleid2 (4.4).

```
[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. 4.4: Установки новых атрибутов и смена владельца файла simpleid2

5. Выполним команды ./simpleid2 и id и убедимся, что полученные данные совпадают (4.5).

```
[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. 4.5: Использование команд ./simpleid2 и id

6. Проделаем то же самое относительно SetGID-бита (4.6).


```
[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. 4.6: Операции с SetGID-битом

7. Создадим и скомпилируем программу readfile.c (4.7).

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

Рис. 4.7: Создание и компиляция программы readfile.c

8. Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог(4.8).

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

Рис. 4.8: Изменение владельца и прав файла readfile.c

9. Проверим, что пользователь guest не может прочитать файл readfile.c. (4.9).

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

Рис. 4.9: Проверка, что пользователь guest не может прочитать файл readfile.c.

10. Сменим у программы readfile владельца и установим SetUID-бит (4.10).

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

Рис. 4.10: Работа с параметрами readfile

11. Проверим, может ли программа readfile прочитать файл readfile.c (4.11).

```
[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. 4.11: Попытка прочитать файл readfile.c программой readfile

12. Проверим, может ли программа readfile прочитать файл /etc/shadow (4.12).

```
[root@info-sec guest]# ./readfile /etc/shadow
root:$6$irGYpLkCLN1K0rj0$g6wUErxhgqnlzil/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. 4.12: Попытка прочитать файл /etc/shadow программой readfile

13. Выясним, установлен ли атрибут Sticky на директории /tmp (4.13).

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

Рис. 4.13: Чтение атрибутов директории /tmp

14. 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» (4.14).

```
[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. 4.14: Чтение атрибутов директории /tmp

15. От пользователя guest2 попробуем прочитать файл /tmp/file01.txt (4.15).

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

Рис. 4.15: Попытка прочтения файла /tmp/file01.txt

16. От пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2 (4.16).

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

Рис. 4.16: Попытка дозаписи в файл /tmp/file01.txt

17. От пользователя guest2 попробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию (4.17).

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

Рис. 4.17: Попытка записи в файл /tmp/file01.txt

18. От пользователя guest2 попробуем удалить файл /tmp/file01.txt (4.18).

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

Рис. 4.18: Попытка удаления файла /tmp/file01.txt

19. От имени суперпользователя снимем атрибут `t` с директории `/tmp`. От пользователя `guest2` проверим, что атрибута `t` у директории `/tmp` нет (4.19).

```
[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. 4.19: Удаление атрибута `t` директории `/tmp`

20. Повторим предыдущие шаги. Теперь мы можем удалить файл (4.20).

```
[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. 4.20: Повторение предыдущих шагов

21. Повысим свои права до суперпользователя и вернем атрибут `t` на директорию `/tmp` (4.20).

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

Рис. 4.21: Возвращение атрибута t директории /tmp

5 Выводы

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.

Список литературы

[1] <https://blog.skillfactory.ru/glossary/operaczionnaya-sistema/>

[2] <https://codechick.io/tutorials/unix-linux/unix-linux-permissions>