

# **Лабораторная работа №5**

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

Ишанова А.И. группа НФИ-02-19

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Теоретическое введение</b>	<b>5</b>
2.1	SetUID . . . . .	5
2.2	Sticky . . . . .	5
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Создание программы . . . . .	6
3.2	Исследование Sticky-бита . . . . .	10
<b>4</b>	<b>Вывод</b>	<b>13</b>
<b>5</b>	<b>Библиография</b>	<b>14</b>

# List of Figures

3.1	Работа в консоли с файлом simpleid.c . . . . .	6
3.2	Содержимое файла simpleid.c . . . . .	6
3.3	Работа в консоли с файлом simpleid2.c . . . . .	7
3.4	Содержимое файла simpleid2.c . . . . .	7
3.5	Изменение прав файла simpleid2 . . . . .	7
3.6	Проверка прав файла simpleid2, его запуск и команда id . . . . .	7
3.7	Выполнения файла с SetGID-битом . . . . .	8
3.8	Содержимое файла readfile.c . . . . .	8
3.9	Создание и компелирование readfile.c . . . . .	8
3.10	Изменение прав файла readfile.c . . . . .	8
3.11	Чтение readfile.c пользователем guest . . . . .	9
3.12	Смена прав у readfile . . . . .	9
3.13	Чтение readfile.c через readfile . . . . .	9
3.14	Чтение /etc/shadow через readfile . . . . .	10
3.15	Создание и изменение прав файла /tmp/file01.txt . . . . .	11
3.16	Взаимдействие с file01.txt пользователем guest2 с Sticky-bit . . .	11
3.17	Взаимдействие с file01.txt пользователем guest2 без Sticky-bit . .	12
3.18	Возвращение Sticky-bit каталогу tmp . . . . .	12

# 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 Теоретическое введение

Дискреционное разграничение доступа — управление доступом субъектов к объектам на основе списков управления доступом или матрицы доступа. Также используются названия дискреционное управление доступом, контролируемое управление доступом и разграничительное управление доступом. [2]

### 2.1 SetUID

setuid и setgid (сокращения от англ. set user ID upon execution — «установка ID пользователя во время выполнения» и англ. set group ID upon execution — «установка ID группы во время выполнения») являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами владельца или группы исполняемого файла. [3]

### 2.2 Sticky

Sticky bit используется в основном для каталогов, чтобы защитить в них файлы. Из такого каталога пользователь может удалить только те файлы, владельцем которых он является. Примером может служить каталог /tmp, в который запись открыта для всех пользователей, но нежелательно удаление чужих файлов. [4]

## 3 Выполнение лабораторной работы

### 3.1 Создание программы

1. Зашли в систему от имени пользователя guest.
2. Создали файл simpleid.c, записали в него программу, скопировали и запустили его. Программа дала те же результаты, что и консольная команда id. (fig. 3.1, fig. 3.2)

```
[guest@aiishanova ~]$ cd dir1
[guest@aiishanova dir1]$ touch simpleid.c
[guest@aiishanova dir1]$ gcc simpleid.c -o simpleid
[guest@aiishanova dir1]$ ./simpleid
uid=1001, gid=1001
[guest@aiishanova dir1]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@aiishanova dir1]$
```

Figure 3.1: Работа в консоли с файлом simpleid.c

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

Figure 3.2: Содержимое файла simpleid.c

3. Создали файл simpleid2.c, записали в него программу, скопировали и запустили его. (fig. 3.3, fig. 3.4)

```
[guest@aiishanova dir1]$ gcc simpleid2.c -o simpleid2
[guest@aiishanova dir1]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Figure 3.3: Работа в консоли с файлом simpleid2.c

```
Open simpleid2.c ~/dir1
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int
6 main()
7 {
8     uid_t real_uid = getuid();
9     uid_t e_uid = geteuid();
10
11     gid_t real_gid = getgid();
12     gid_t e_gid = getegid();
13
14     printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
15     printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
16     return 0;
17 }
```

Figure 3.4: Содержимое файла simpleid2.c

4. Изменили права файла simpleid2 от имени суперпользователя. (fig. 3.5)

```
[root@aiishanova guest]# chown root:guest /home/guest/dir1/simpleid2
[root@aiishanova guest]# chmod u+s /home/guest/dir1/simpleid2
```

Figure 3.5: Изменение прав файла simpleid2

5. Выполнили проверку установки правил. Запустили simpleid2 и id. Получили одинаковы результаты с id=0. (fig. 3.6)

```
[guest@aiishanova dir1]$ su
Password:
[root@aiishanova dir1]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct  8 17:24 simpleid2
[root@aiishanova dir1]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@aiishanova dir1]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
[root@aiishanova dir1]#
```

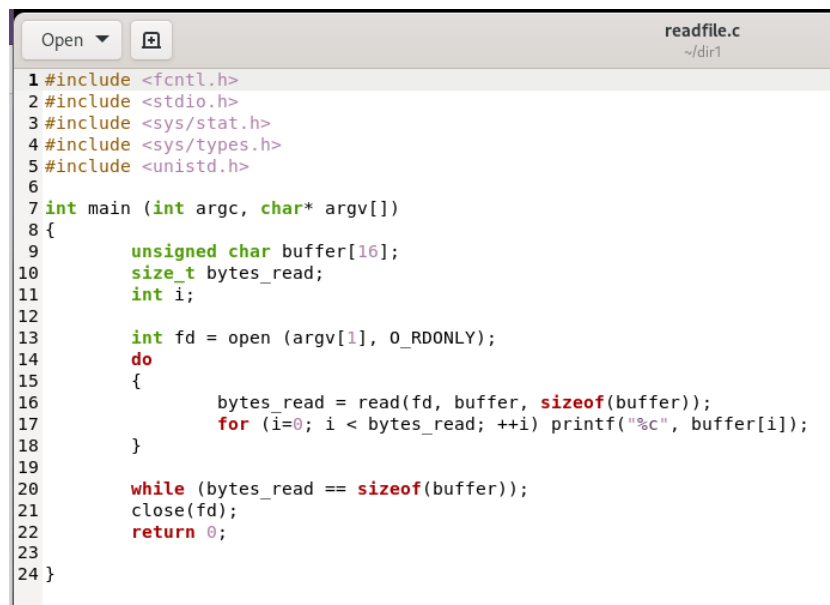
Figure 3.6: Проверка прав файла simpleid2, его запуск и команда id

6. Повторили п.5 для SetGID-бита. (fig. 3.7)

```
[root@aiishanova dir1]# chmod g+s /home/guest/dir1/simpleid2
[root@aiishanova dir1]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 26008 Oct  8 17:24 simpleid2
[root@aiishanova dir1]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@aiishanova dir1]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned t:s0-s0:c0.c1023
[root@aiishanova dir1]#
```

Figure 3.7: Выполнения файла с SetGID-битом

7. Создали программу readfile.c и откомпилировали ее. (fig. 3.8, fig. 3.9)



```
1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <sys/stat.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6
7 int main (int argc, char* argv[])
8 {
9     unsigned char buffer[16];
10    size_t bytes_read;
11    int i;
12
13    int fd = open (argv[1], O_RDONLY);
14    do
15    {
16        bytes_read = read(fd, buffer, sizeof(buffer));
17        for (i=0; i < bytes_read; ++i) printf("%c", buffer[i]);
18    }
19
20    while (bytes_read == sizeof(buffer));
21    close(fd);
22    return 0;
23
24 }
```

Figure 3.8: Содержимое файла readfile.c

```
[root@aiishanova dir1]# touch readfile.c
[root@aiishanova dir1]# gcc readfile.c -o readfile
```

Figure 3.9: Создание и компелирование readfile.c

8. Изменили права так, чтобы только суперпользователь (root) мог прочитать readfile.c, а guest не мог. (fig. 3.10)

```
[root@aiishanova guest]# chown root:guest /home/guest/dir1/readfile.c
[root@aiishanova guest]# chmod 700 /home/guest/dir1/readfile.c
```

Figure 3.10: Изменение прав файла readfile.c



9. Проверили, что guest не может прочитать файл. (fig. 3.11)

```
Completion terminated.  
[guest@aiishanova dir1]$ cat readfile.c  
cat: readfile.c: Permission denied  
[guest@aiishanova dir1]$
```

Figure 3.11: Чтение readfile.c пользователем guest

10. Сменили у программы readfile владельца и установили SetU'D-бит. (fig. 3.12)

```
[root@aiishanova guest]# chown root /home/guest/dir1/readfile  
[root@aiishanova guest]# chmod u+s /home/guest/dir1/readfile
```

Figure 3.12: Смена прав у readfile

11. Считали программой readfile readfile.c и /etc/shadow. (fig. 3.13, fig. 3.14)

```
[root@aiishanova dir1]# ./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];  
    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;  
}  
[root@aiishanova dir1]#
```

Figure 3.13: Чтение readfile.c через readfile

```
[root@aiishanova dir1]# ./readfile /etc/shadow
root:$6$Z9FZzhrF5K8PIvd$M15WSCitCKRYc15FketT0.8yq0qlwVh./BAe05e6NlFzz0BTkv5
0wdw5wfBTEXjegWbh4D/eHh6ToDdL3/err0::0:99999:7:::
bin:!:19123:0:99999:7:::
daemon:!:19123:0:99999:7:::
adm:!:19123:0:99999:7:::
lp:!:19123:0:99999:7:::
sync:!:19123:0:99999:7:::
shutdown:!:19123:0:99999:7:::
halt:!:19123:0:99999:7:::
mail:!:19123:0:99999:7:::
operator:!:19123:0:99999:7:::
games:!:19123:0:99999:7:::
ftp:!:19123:0:99999:7:::
nobody:!:19123:0:99999:7:::
systemd-coredump:!!:19245::::::
dbus:!!:19245::::::
polkitd:!!:19245::::::
rtkit:!!:19245::::::
sssd:!!:19245::::::
avahi:!!:19245::::::
pipewire:!!:19245::::::
libstoragemgmt:!!:19245::::::
tss:!!:19245::::::
geoclue:!!:19245::::::
cockpit-ws:!!:19245::::::
cockpit-wsinstance:!!:19245::::::
setroubleshoot:!!:19245::::::
flatpak:!!:19245::::::
colord:!!:19245::::::
clevis:!!:19245::::::
gdm:!!:19245::::::
systemd-oom:!*:19245::::::
pesign:!!:19245::::::
gnome-initial-setup:!!:19245::::::
sshd:!!:19245::::::
chrony:!!:19245::::::
dnsmasq:!!:19245::::::
tcpdump:!!:19245::::::
aiishanova:$6$GGuyK9WoP5ha/h6j$/Kv0vEj75G1qVP1PnQRYLKSkbERp.dXf.7KycD.dzw.WP
AiAtXNCvTqGXZ1Aysc21imK/j08Js03uP5Z0IrrS1::0:99999:7:::
guest:$6$b8hDYG8LZ2GD1yfe$EYKtllUKWRMdmGofbISwy/A3ejlifQ00SccejGtYhMQ.6k8zNw
WdLjdHh19ea5Za.p8RYsqy5CqXEcx7kzhNf/:19252:0:99999:7:::
```

Figure 3.14: Чтение /etc/shadow через readfile

## 3.2 Исследование Sticky-бита

1. Проверили установлены ли на директории tmp атрибут Sticky. От имени пользователя guest создали file01.txt в директории /tmp со словом test. Просмотрели атрибуты у файла и разрешили чтение и запись для категории пользователей «все остальные». (fig. 3.15)

```
[guest@aiishanova ~]$ ls -l / |grep tmp
drwxrwxrwt. 16 root root 4096 Oct  8 17:43 tmp
[guest@aiishanova ~]$ echo "test" > /tmp/file01.txt
[guest@aiishanova ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Oct  8 17:51 /tmp/file01.txt
[guest@aiishanova ~]$ chmod o+rw /tmp/file01.txt
[guest@aiishanova ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Oct  8 17:51 /tmp/file01.txt
[guest@aiishanova ~]$
```

Figure 3.15: Создание и изменение прав файла /tmp/file01.txt

2. От имени пользователя guest2 попробовали прочитать, дозаписать, переписать и удалить файл file01.txt. (fig. 3.16)

```
[guest@aiishanova ~]$ su guest2
Password:
[guest2@aiishanova guest]$ cat /tmp/file01.txt
test
[guest2@aiishanova guest]$ echo "test2" >> /tmp/file01.txt
[guest2@aiishanova guest]$ cat /tmp/file01.txt
test
test2
[guest2@aiishanova guest]$ echo "test3" > /tmp/file01.txt
[guest2@aiishanova guest]$ cat /tmp/file01.txt
test3
[guest2@aiishanova guest]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
```

Figure 3.16: Взаимодействие с file01.txt пользователем guest2 с Sticky-bit

3. Суперпользователем сняли Sticky-bit с каталога tmp. Повторили действия с файлом из п.2. (fig. 3.17)

```

[guest2@aiishanova guest]$ su -
Password:
[root@aiishanova ~]# chmod -t /tmp
[root@aiishanova ~]# exit
logout
[guest2@aiishanova guest]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 Oct  8 18:01 tmp
[guest2@aiishanova guest]$ cat /tmp/file01.txt
test3
[guest2@aiishanova guest]$ echo "test2" >> /tmp/file01.txt
[guest2@aiishanova guest]$ cat /tmp/file01.txt
test3
test2
[guest2@aiishanova guest]$ echo "test3" > /tmp/file01.txt
[guest2@aiishanova guest]$ cat /tmp/file01.txt
test3
[guest2@aiishanova guest]$ rm /tmp/file01.txt
[guest2@aiishanova guest]$

```

Figure 3.17: Взаимодействие с file01.txt пользователем guest2 без Sticky-bit

4. Вернули каталогу tmp Sticky-bit суперпользователем. (fig. 3.18)

```

[guest2@aiishanova guest]$ su -
Password:
[root@aiishanova ~]# chmod +t /tmp
[root@aiishanova ~]# exit
logout
[guest2@aiishanova guest]$

```

Figure 3.18: Возвращение Sticky-bit каталогу tmp

## 4 Вывод

В ходе выполнения лабораторной работы были опробованы действия на практике SetUID- и Sticky-битов и рассмотрен механизм смены идентификатора процессов пользователей.

## 5 Библиография

1. Методические материалы курса.
2. Wikipedia: Избирательное управление доступом. (URL: [https://ru.wikipedia.org/wiki/%D0%9A%D0%B8%D0%BE%D0%BB%D0%BD\\_%D0%A6%D0%BC%D0%BA%D0%B7%D0%BF](https://ru.wikipedia.org/wiki/%D0%9A%D0%B8%D0%BE%D0%BB%D0%BD_%D0%A6%D0%BC%D0%BA%D0%B7%D0%BF))
3. Wikipedia: suid (URL: <https://ru.wikipedia.org/wiki/Suid>)
4. Wikipedia: Sticky bit (URL: [https://ru.wikipedia.org/wiki/Sticky\\_bit](https://ru.wikipedia.org/wiki/Sticky_bit))