

# **Отчет по лабораторной работе №5**

**по дисциплине: Информационная безопасность**

Го Чаопэн

# Содержание

<b>1</b>	<b>Цели работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>5</b>	<b>Выводы</b>	<b>14</b>
<b>6</b>	<b>Список литературы</b>	<b>15</b>

## Список иллюстраций

4.1	Использование команд ./simpleid и id . . . . .	7
4.2	Запуск программы simpleid2 . . . . .	8
4.3	Установки новых атрибутов и смена владельца файла simpleid2 . . . . .	8
4.4	Использование команд ./simpleid2 . . . . .	9
4.5	Операции с SetGID-битом . . . . .	9
4.6	Изменение владельца и прав файла readfile.c . . . . .	9
4.7	Работа с параметрами readfile . . . . .	10
4.8	Попытка прочитать файл readfile.c программой readfile . . . . .	10
4.9	Попытка прочитать файл /etc/shadow программой readfile . . . . .	11
4.10	Чтение атрибутов директории /tmp . . . . .	11
4.11	Создание файла /tmp/file01.txt . . . . .	12
4.12	Работа с файлом /tmp/file01.txt . . . . .	12
4.13	Удаление атрибута t директории /tmp . . . . .	13

# 1 Цели 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. Выполним команды ./simpleid и id и убедимся, что полученные данные совпадают (4.1).

```
[guest@guo ~]$ touch simpleid.c
[guest@guo ~]$ nano simpleid.c
[guest@guo ~]$ gcc simpleid.c -o simpleid
[guest@guo ~]$ ./simpleid
uid=1001, gid=1001
[guest@guo ~]$ id
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@guo ~]$
```

Рис. 4.1: Использование команд ./simpleid и id

Код программы 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;
}
```

2. Усложним программу и запишем ее в файл simpleid2.c. Запустим получившуюся программу (4.2).

```
[guest@guo lab5]$ ./simpleid2
uid=1001, gid=1001
[guest@guo lab5]$
```

Рис. 4.2: Запуск программы simpleid2

Код программы simpleid2.c:

```
#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;
}
```

3. От имени суперпользователя установим новые атрибуты и сменим владельца файла simpleid2 (4.3).

```
[guo@guo ~]$ sudo su
[sudo] пароль для guo:
[root@guo guo]# chown root:guest /home/guest/lab5/simpleid2
[root@guo guo]# chmod u+s /home/guest/lab5/simpleid2
[root@guo guo]#
```

Рис. 4.3: Установки новых атрибутов и смена владельца файла simpleid2



4. Выполним команду ./simpleid2 (4.4).

```
[root@guo lab5]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@guo lab5]#
```

Рис. 4.4: Использование команд ./simpleid2

5. Проделаем то же самое относительно SetGID-бита (4.5).

```
[root@guo lab5]# chmod g+s simpleid2
[root@guo lab5]# ls -l simpleid2
-rwxr-sr-x. 1 root root 26064 окт  7 17:17 simpleid2
[root@guo lab5]# exit
exit
[guest@guo lab5]$ ./simpleid2
e_uid=1001, e_gid=0
real_uid=1001, real_gid=1001
[guest@guo lab5]$ id
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@guo lab5]$
```

Рис. 4.5: Операции с SetGID-битом

6. Создадим и скомпилируем программу readfile.c. Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (4.6).

```
[guest@guo lab5]$ nano readfile.c
[guest@guo lab5]$ gcc readfile.c -o readfile
[guest@guo lab5]$ su
Пароль:
[root@guo lab5]# chown root:guest readfile.c
[root@guo lab5]# chmod 700 readfile.c
[root@guo lab5]# exit
exit
[guest@guo lab5]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@guo lab5]$
```

Рис. 4.6: Изменение владельца и прав файла readfile.c

Пользователь guest не может прочитать файл readfile.c

7. Сменим у программы readfile владельца и установим SetUID-бит (4.7).

```
[guest@guo lab5]$ su
Пароль:
[root@guo lab5]# chown root:guest readfile
[root@guo lab5]# chmodd u+s readfile
bash: chmodd: command not found...
Similar command is: 'chmod'
[root@guo lab5]# chmod u+s readfile
[root@guo lab5]#
```

Рис. 4.7: Работа с параметрами readfile

8. Проверим, может ли программа readfile прочитать файл readfile.c (4.8).

```
[guest@guo lab5]$ ./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;
}
[guest@guo lab5]$ ./readfile /etc/shadow
```

Рис. 4.8: Попытка прочитать файл readfile.c программой readfile

9. Проверим, может ли программа readfile прочитать файл /etc/shadow (4.9).

```
[guest@guo lab5]$ ./readfile /etc/shadow
root:$6$SL6WRXb0EvAP6eX$SGgp.pI8mf8em1By0K0f3pWdocBpbxCBvae6yIpldAItC90Ir76Bd2EBu0TzlefDjeQG9Pmdic5jd0gJVBxJy...: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:::19607:7:::
dbus:::19607:7:::
polkitd:::19607:7:::
avahi:::19607:7:::
rtkit:::19607:7:::
sssd:::19607:7:::
pipewire:::19607:7:::
libstoragemgmt:::19607:7:::
systemd-oom:::19607:7:::
tss:::19607:7:::
geoclue:::19607:7:::
cockpit-ws:::19607:7:::
cockpit-wsinstance:::19607:7:::
flatpak:::19607:7:::
colord:::19607:7:::
cleviis:::19607:7:::
setroubleshoot:::19607:7:::
gdm:::19607:7:::
pesign:::19607:7:::
gnome-initial-setup:::19607:7:::
sshd:::19607:7:::
chrony:::19607:7:::
dnsmasq:::19607:7:::
tcpdump:::19607:7:::
guo:$6$E/5LtU2Hd2qzEoaC$.khwrdCtCjm60kRWz/cFLkT8qLxshD0twOh.ZR/u2f0aApP3i3ac/iNEllGT2PUk8NPcxRrDeXJFmhwsmlXl2/:0:99999:7:::
guest:$6$e2i8uW2VlFrC3i$W0WNbhysSV0q3YJw0VzS85kPOxfadi6Lx9QFMW6yRoDgaVDWUDN3xLU2NGxLUMckFPJYUsd5embsgmD1FG/hf1:19616:0:99999:7:::
guest2:$6$03lEx2U0LhtWAGW$XubnoKpMtp.qT00z3UHLd6IiUG/eVS0z6HBClJ5FWLnRxxMSszb6he.IncyNoi75BGtdmKPKoXB559lZKn1CE1:19621:0:99999:7:::
[guest@guo lab5]$
```

Рис. 4.9: Попытка прочитать файл /etc/shadow программой readfile

10. Выясним, установлен ли атрибут Sticky на директории /tmp (4.10).

```
[guest@guo lab5]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 окт  7 17:31 tmp
[guest@guo lab5]$
```

Рис. 4.10: Чтение атрибутов директории /tmp

11. 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» (4.11).

```
[guest@guo lab5]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 окт  7 17:31 tmp
[guest@guo lab5]$ echo "test" > /tmp/file01.txt
[guest@guo lab5]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 окт  7 17:35 /tmp/file01.txt
[guest@guo lab5]$ chmod o+rw /tmp/file01.txt
[guest@guo lab5]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 окт  7 17:35 /tmp/file01.txt
[guest@guo lab5]$
```

Рис. 4.11: Создание файла /tmp/file01.txt


12. От пользователя guest2 попробуем прочитать файл /tmp/file01.txt. Далее попробуем дозаписать в файл /tmp/file01.txt слово test2, записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию. После этого попробуем удалить данный файл (4.12).

```
[guest2@guo lab5]$ cat /tmp/file01.txt
test
[guest2@guo lab5]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@guo lab5]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@guo lab5]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@guo lab5]$
```

Рис. 4.12: Работа с файлом /tmp/file01.txt

Пользователь guest2 принадлежит группе guest, поэтому у него нет доступа к вышеописанным действиям, так как у группы нет права доступа на запись для данного файла.

13. От имени суперпользователя снимем атрибут t с директории /tmp. От пользователя guest2 проверим, что атрибута t у директории /tmp нет. Повторим предыдущие шаги. Теперь мы можем удалить файл. (4.13).



```
[guest2@guo lab5]$ su -
Пароль:
[root@guo ~]# chmod -t /tmp
[root@guo ~]# exit
Выход
[guest2@guo lab5]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 окт  7 18:01 tmp
[guest2@guo lab5]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@guo lab5]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@guo lab5]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
[guest2@guo lab5]$
```

Рис. 4.13: Удаление 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.

## 6 Список литературы

1. Операционные системы [Электронный ресурс]. URL: <https://softline.tm/solutions/programmnoe-obespechenie/operating-system>.
2. Права доступа [Электронный ресурс]. URL: <https://w.wiki/7UBB>.