

# Лабораторная работа № 5

Соболевский Денис Андреевич

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 влияние дополнительных атрибутов.
- 2 Исследовать Sticky-бит.

# Ход работы

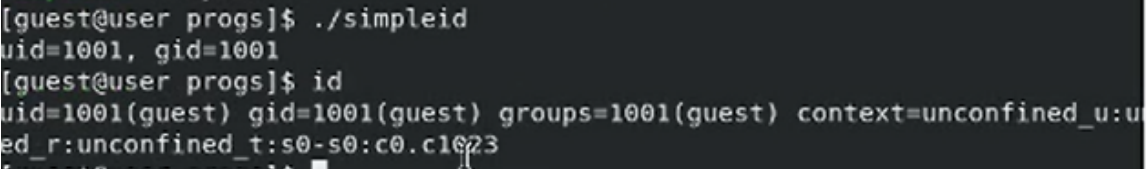
От имени пользователя guest создадим программу simpleid.c, скомпилируем ее и убедимся, что файл программы создан.



The screenshot shows a code editor window with a title bar that includes a button labeled "Open" with a dropdown arrow, a file icon, and the filename "simpleid.c" with the path "~/progs". The code is written in C and uses syntax highlighting: preprocessor directives are magenta, keywords are green, identifiers and literals are black, and the return statement is red. The code is as follows:

```
#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;
}
```

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

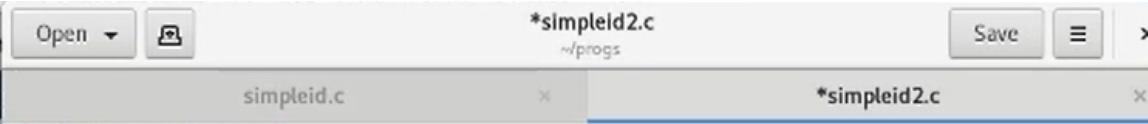


```
[guest@user progs]$ ./simpleid
uid=1001, gid=1001
[guest@user progs]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 1: Выполнение команд `./simpleid` и `id`

# Ход работы

Усложним программу, добавив вывод действительных идентификаторов.



The screenshot shows a code editor window with two tabs: 'simpleid.c' and '\*simpleid2.c'. The active tab is '\*simpleid2.c', which contains the following C code:

```
#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;
}
```

От имени суперпользователя выполним команды.

```
[root@user progs]# chown root:guest simpleid2  
[root@user progs]# cchmod u+s simpleid2  
bash: cchmod: command not found...  
[root@user progs]# chmod u+s simpleid2
```

Рис. 3: Установка новых атрибутов и смена владельца файла simpleid2

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

A terminal window with a dark background and light gray text. The first command executed is `./simpleid2`, which outputs `e_uid=0, e_gid=0` and `real_uid=0, real_gid=0`. The second command is `id`, which outputs `uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023`.

```
[root@user progs]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@user progs]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 4: Использование команд `./simpleid2` и `id`

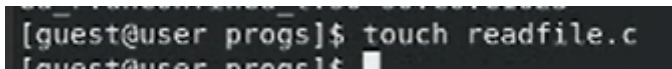


Выполним проверку правильности установки новых атрибутов.

```
[root@user progs]# ls -l simpleid2  
-rwsrwsr-x. 1 root guest 13064 Oct  7 12:32 simpleid2  
[root@user progs]# exit
```

Рис. 5: Выполнение команды `ls -l simpleid2`

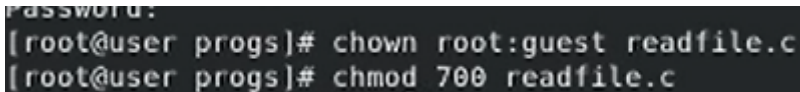
Создадим и скомпилируем программу readfile.c.

A terminal window with a dark background. The prompt is [guest@user progs]\$. The command touch readfile.c has been entered and executed. The next line shows the prompt [guest@user progs]\$ with a cursor at the end.

```
[guest@user progs]$ touch readfile.c  
[guest@user progs]$
```

Рис. 6: Создание программы readfile.c

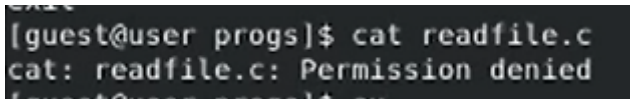
Сменим владельца у файла `readfile.c` и изменим права так, чтобы только суперпользователь (`root`) мог прочитать его, а `guest` не мог.

A terminal window with a dark background and light-colored text. The text shows a password prompt, followed by two commands executed as root in a directory named 'progs'. The first command is 'chown root:guest readfile.c' and the second is 'chmod 700 readfile.c'.

```
password:  
[root@user progs]# chown root:guest readfile.c  
[root@user progs]# chmod 700 readfile.c
```

Рис. 7: 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@user progs]\$. The command entered is cat readfile.c. The output is cat: readfile.c: Permission denied. The next line shows the prompt [guest@user progs]\$ again.

```
[guest@user progs]$ cat readfile.c  
cat: readfile.c: Permission denied  
[guest@user progs]$
```

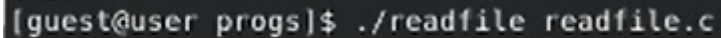
Рис. 8: Проверка, что пользователь guest не может прочитать файл readfile.c

Сменим владельца и установим SetUID-бит.

```
password:  
[root@user progs]# chown root:guest readfile  
[root@user progs]# chmod u+s readfile
```

Рис. 9: Смена прав файла

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

A screenshot of a terminal window with a dark background. The text '[guest@user progs]\$ ./readfile readfile.c' is displayed in a light-colored, monospaced font. The prompt character is a dollar sign, and the command consists of the program name followed by the filename.

```
[guest@user progs]$ ./readfile readfile.c
```

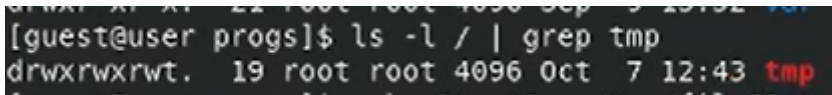
Рис. 10: Чтение файла readfile.c

## Ход работы

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

```
saslauthd:!!:19609:::~  
dnsmasq:!!:19609:::~  
radvd:!!:19609:::~  
sssd:!!:19609:::~  
clevis:!!:19609:::~  
cockpit-ws:!!:19609:::~  
cockpit-wsinstance:!!:19609:::~  
flatpak:!!:19609:::~  
colord:!!:19609:::~  
rpcuser:!!:19609:::~  
gdm:!!:19609:::~  
gnome-initial-setup:!!:19609:::~  
pesign:!!:19609:::~  
sshd:!!:19609:::~  
rngd:!!:19609:::~  
tcpdump:!!:19609:::~  
dasobolevskiy:$6$sr2S3AypLds/Taj5$NbcWgUvi.vrDoHiTCXGZL5ffb6q9Q4IJZzoYfLLjFrL0TA  
THiY8qIu2HFgiLFektEWhBt4hDzFcEBRztSnr8J.:19609:0:99999:7::~  
vboxadd:!!:19609:::~
```

Выясним, установлен ли атрибут Sticky на директории /tmp.

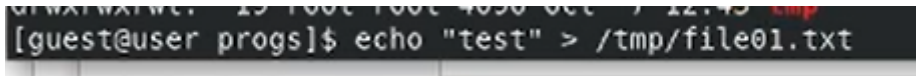


```
[guest@user progs]$ ls -l / | grep tmp
drwxrwxrwt. 19 root root 4096 Oct  7 12:43 tmp
```

Рис. 12: Выполнение команды `ls -l / | grep tmp`



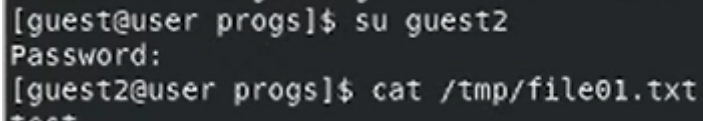
От имени пользователя guest создадим файл file01.txt в директории /tmp.

A screenshot of a terminal window with a dark background. The prompt is [guest@user progs]\$ and the command being entered is echo "test" > /tmp/file01.txt. The word 'tmp' in the path is highlighted in red. There is some faint, illegible text visible in the background of the terminal window.

```
[guest@user progs]$ echo "test" > /tmp/file01.txt
```

Рис. 13: Создание файла file01.txt

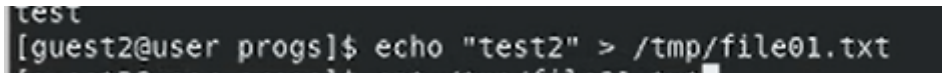
От пользователя guest2 попробуем прочитать файл file01.txt.



```
[guest@user progs]$ su guest2
Password:
[guest2@user progs]$ cat /tmp/file01.txt
test
```

Рис. 14: Чтение файла file01.txt

От пользователя `guest2` попробуем дозаписать файл `file01.txt`.

A terminal window with a dark background. The prompt is [guest2@user progs]\$. The command entered is echo "test2" > /tmp/file01.txt. The output of the command is test2.

```
test  
[guest2@user progs]$ echo "test2" > /tmp/file01.txt  
test2
```

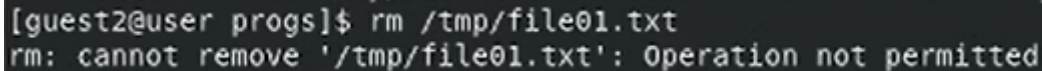
Рис. 15: Дозапись в файл `/tmp/file01.txt`

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

```
test3  
[guest2@user progs]$ echo "test3" > /tmp/file01.txt  
[guest2@user progs]$ cat /tmp/file01.txt  
test3
```

Рис. 16: Перезапись в файле /tmp/file01.txt

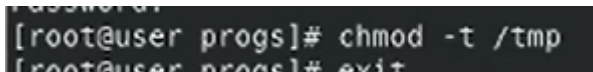
От пользователя guest2 попробуем удалить файл file01.txt.

A terminal window with a dark background and light gray text. The prompt is [guest2@user progs]\$. The command entered is rm /tmp/file01.txt. The output is rm: cannot remove '/tmp/file01.txt': Operation not permitted.

```
[guest2@user progs]$ rm /tmp/file01.txt  
rm: cannot remove '/tmp/file01.txt': Operation not permitted
```

Рис. 17: Удаление файла /tmp/file01.txt

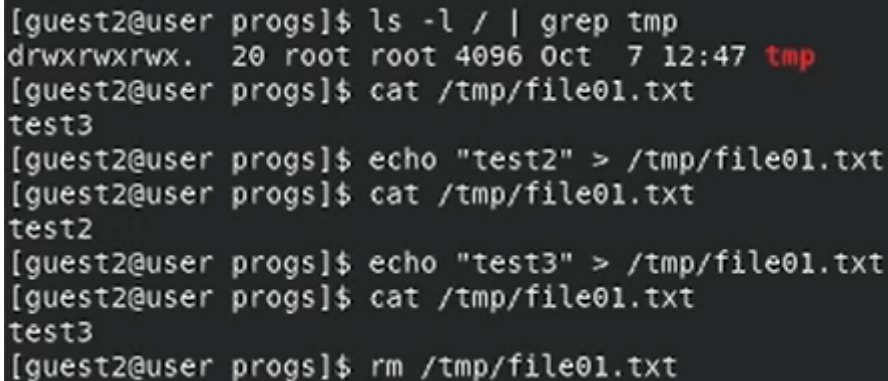
От имени суперпользователя снимем атрибут `t` с директории `/tmp`..

A terminal window screenshot showing a root shell session. The prompt is [root@user progs]#. The command entered is chmod -t /tmp. The output is [root@user progs]#. The command entered is exit. The output is [root@user progs]#. The terminal background is dark, and the text is light gray.

```
[root@user progs]# chmod -t /tmp
[root@user progs]# exit
```

Рис. 18: Удаление атрибута `t`

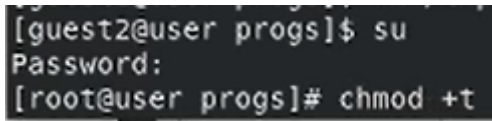
Повторим предыдущие шаги. Теперь файл удален успешно.

A terminal window showing a series of commands and their outputs. The user is in a shell as 'guest2' in the directory 'progs'. The commands and outputs are: 1. 'ls -l / | grep tmp' returns 'drwxrwxrwx. 20 root root 4096 Oct 7 12:47 tmp' (the word 'tmp' is highlighted in red). 2. 'cat /tmp/file01.txt' returns 'test3'. 3. 'echo "test2" > /tmp/file01.txt' returns nothing. 4. 'cat /tmp/file01.txt' returns 'test2'. 5. 'echo "test3" > /tmp/file01.txt' returns nothing. 6. 'cat /tmp/file01.txt' returns 'test3'. 7. 'rm /tmp/file01.txt' returns nothing.

```
[guest2@user progs]$ ls -l / | grep tmp
drwxrwxrwx. 20 root root 4096 Oct 7 12:47 tmp
[guest2@user progs]$ cat /tmp/file01.txt
test3
[guest2@user progs]$ echo "test2" > /tmp/file01.txt
[guest2@user progs]$ cat /tmp/file01.txt
test2
[guest2@user progs]$ echo "test3" > /tmp/file01.txt
[guest2@user progs]$ cat /tmp/file01.txt
test3
[guest2@user progs]$ rm /tmp/file01.txt
```

Рис. 19: Повторение предыдущих шагов

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



```
[guest2@user progs]$ su  
Password:  
[root@user progs]# chmod +t
```

A terminal window with a dark background. The first line shows a user prompt '[guest2@user progs]\$' followed by the command 'su'. The second line shows the prompt 'Password:'. The third line shows the root prompt '[root@user progs]#' followed by the command 'chmod +t'.

Рис. 20: Повышение прав и возвращение атрибута



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.