

# **Основы информационной безопасности**

**Лабораторная работа № 5. Исследование влияния дополнительных атрибутов**

Демидова Екатерина Алексеевна

# Содержание

1	Цель работы	4
2	Теоретические сведения	5
3	Выполнение лабораторной работы	6
4	Выводы	13
	Список литературы	14

## Список иллюстраций

3.1	Подготовка лабораторного стенда . . . . .	6
3.2	Текст программы simpleid.c . . . . .	7
3.3	Запуск программы simpleid . . . . .	7
3.4	Текст программы simpleid2.c . . . . .	8
3.5	Запуск программы simpleid2 . . . . .	8
3.6	Изменение владельца и запуск программы simpleid2 с установленным SetUID-битом . . . . .	9
3.7	Запуск программы simpleid2 с установленным SetGID-битом . . . . .	9
3.8	Текст программы readfile.c . . . . .	10
3.9	Изменение владельца и прав файла readfile.c . . . . .	11
3.10	Установка SetUID-бита на исполняемый файл readfile и проверка прав . . . . .	11
3.11	Подключение образа диска дополнений . . . . .	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 Теоретические сведения

При работе с командой `chmod` важно понимать основные права доступа, которые назначают файлам или каталогам. В Linux используется три основных типа прав доступа[1]:

- Чтение (Read) — обозначается буквой «r». Предоставляет возможность просматривать содержимое файла или каталога.
- Запись (Write) — обозначается буквой «w». Позволяет создавать, изменять и удалять файлы внутри каталога, а также изменять содержимое файла.
- Выполнение (Execute) — обозначается буквой «x». Дает разрешение на выполнение файла или на вход в каталог.

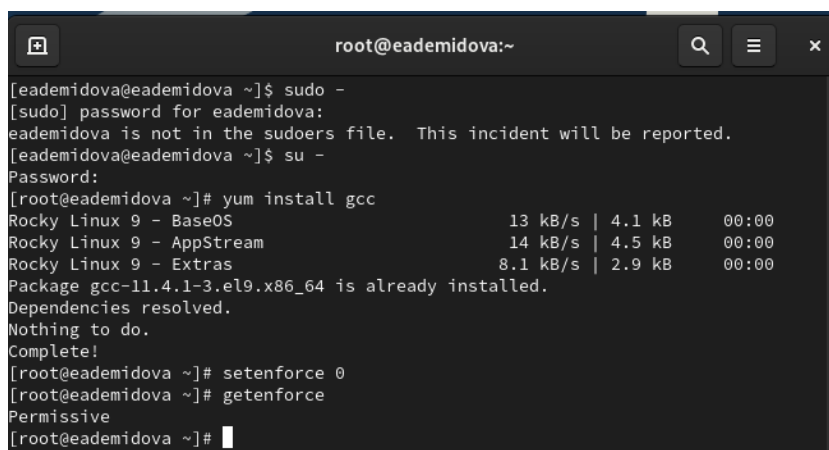
Каждый из указанных выше типов прав доступа может быть назначен трем группам пользователей:

- Владелец (Owner) — пользователь, который является владельцем файла или каталога.
- Группа (Group) — группа пользователей, к которой принадлежит файл или каталог.
- Остальные пользователи (Others) — все остальные пользователи системы.

Комбинация этих базовых прав доступа для каждой из групп пользователей определяет полный набор прав доступа для файла или каталога.

### 3 Выполнение лабораторной работы

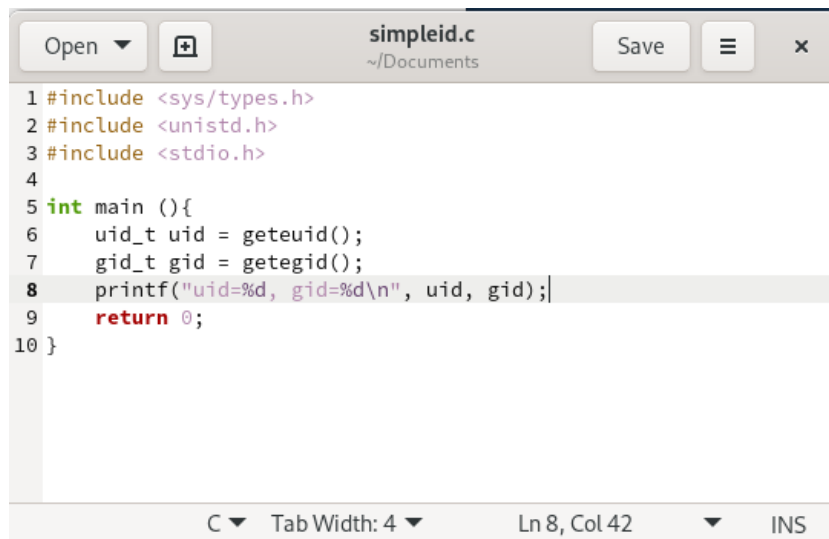
Проверим установлен ли компилятор gcc, а также отключим SELinux(рис. 3.1)



```
root@eademidova:~  
[eademidova@eademidova ~]$ sudo -  
[sudo] password for eademidova:  
eademidova is not in the sudoers file. This incident will be reported.  
[eademidova@eademidova ~]$ su -  
Password:  
[root@eademidova ~]# yum install gcc  
Rocky Linux 9 - BaseOS           13 kB/s | 4.1 kB    00:00  
Rocky Linux 9 - AppStream        14 kB/s | 4.5 kB    00:00  
Rocky Linux 9 - Extras           8.1 kB/s | 2.9 kB    00:00  
Package gcc-11.4.1-3.el9.x86_64 is already installed.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[root@eademidova ~]# setenforce 0  
[root@eademidova ~]# getenforce  
Permissive  
[root@eademidova ~]#
```

Рис. 3.1: Подготовка лабораторного стенда

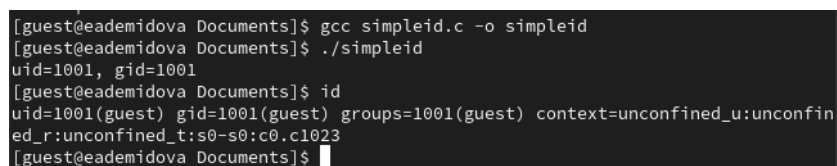
Войдем в систему от имени пользователя guest и создадим программу simpleid.c, которая выводит идентификатор пользователя и группы(рис. 3.2)



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

Рис. 3.2: Текст программы simpleid.c

Теперь скомпилируем программу с помощью gcc, затем, запустив её, увидим, что она выводит идентификаторы пользователя и группы 1001 и 1001 для guest, что совпадает с выводом команды id(рис. 3.3)



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

Рис. 3.3: Запуск программы simpleid

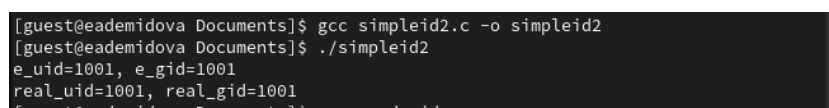
Усложним программу, добавив вывод действительных идентификаторов(рис. 3.4).



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

Рис. 3.4: Текст программы simpleid2.c

Теперь скомпилируем программу с помощью gcc, затем, запустив её, увидим, что она выводит идентификаторы пользователя и группы 1001 и 1001 для guest, что совпадает с выводом команды id(рис. 3.5).



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

Рис. 3.5: Запуск программы simpleid2

От имени суперпользователя изменим владельца файла /home/guest/simpleid2 и установим SetUID-бит. Проверим корректность установленных прав и опять запустим simpleid2(рис. 3.6).



```
root@eademidova:~  
[eademidova@eademidova ~]$ su -  
Password:  
[root@eademidova ~]# chown root:guest /home/guest/Documents/simpleid2  
[root@eademidova ~]# chmod u+s /home/guest/Documents/simpleid2  
[root@eademidova ~]# ls -l /home/guest/Documents/simpleid2  
-rwsr-xr-x. 1 root guest 24488 Jul  6 13:07 /home/guest/Documents/simpleid2  
[root@eademidova ~]# exit  
logout  
[eademidova@eademidova ~]$ su - guest  
Password:  
[guest@eademidova ~]$ cd Documents/  
[guest@eademidova Documents]$ ./simpleid2  
e_uid=0, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@eademidova Documents]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@eademidova Documents]$ su - eademidova  
Password:  
[eademidova@eademidova ~]$ su /home/guest/Documents/simpleid2  
su: user /home/guest/Documents/simpleid2 does not exist or the user entry does not contain all the required fields  
[eademidova@eademidova ~]$ su -  
Password:  
[root@eademidova ~]# /home/guest/Documents/simpleid2  
e_uid=0, e_gid=0  
real_uid=0, real_gid=0  
[root@eademidova ~]# id  
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[root@eademidova ~]#
```

Рис. 3.6: Изменение владельца и запуск программы simpleid2 с установленным SetUID-битом

Прделаем аналогичные действия относительно SetGID-бита(рис. 3.7):

```
[guest@eademidova Documents]$ su -  
Password:  
[root@eademidova ~]# chmod u-s /home/guest/Documents/simpleid2  
[root@eademidova ~]# chmod g+s /home/guest/Documents/simpleid2  
[root@eademidova ~]# exit  
logout  
[guest@eademidova Documents]$ ls -l simpleid2  
-rwxr-sr-x. 1 root guest 24488 Jul  6 13:07 simpleid2  
[guest@eademidova Documents]$ ./simpleid2  
e_uid=1001, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@eademidova Documents]$ ud  
bash: ud: command not found...  
Similar command is: 'du'  
[guest@eademidova Documents]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@eademidova Documents]$
```

Рис. 3.7: Запуск программы simpleid2 с установленным SetGID-битом

Создадим программу для чтения файлов readfile.c(рис. 3.8):

```

GNU nano 5.6.1 /home/guest/Documents/readfile.c
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.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;
}

```

Рис. 3.8: Текст программы readfile.c

Скомпилируем её и сменим владельца у файла с текстом программы, затем изменим права так, чтобы только суперпользователь (root) мог прочитать его, и проверим корректность настроек(рис. 3.9):

```
guest@eademidova:~$ gcc readfile.c -o readfile
[guest@eademidova Documents]$ ./readfile
[guest@eademidova Documents]$ cat readfile.c
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>

int main(int argc, char* argv[]){
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    int fd = open(argv[1], O_RDONLY);
    close(fd);
    return 0;
}
[guest@eademidova Documents]$ su -
Password:
[root@eademidova ~]# chown root:guest /home/guest/Documents/readfile.c
[root@eademidova ~]# chmod 700 /home/guest/Documents/readfile.c
[root@eademidova ~]# cat /home/guest/Documents/readfile.c
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>

int main(int argc, char* argv[]){
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    int fd = open(argv[1], O_RDONLY);
    close(fd);
    return 0;
}
[root@eademidova ~]# exit
logout
[guest@eademidova Documents]$ cat readfile.c
cat: readfile.c: Permission denied
```

Рис. 3.9: Изменение владельца и прав файла readfile.c

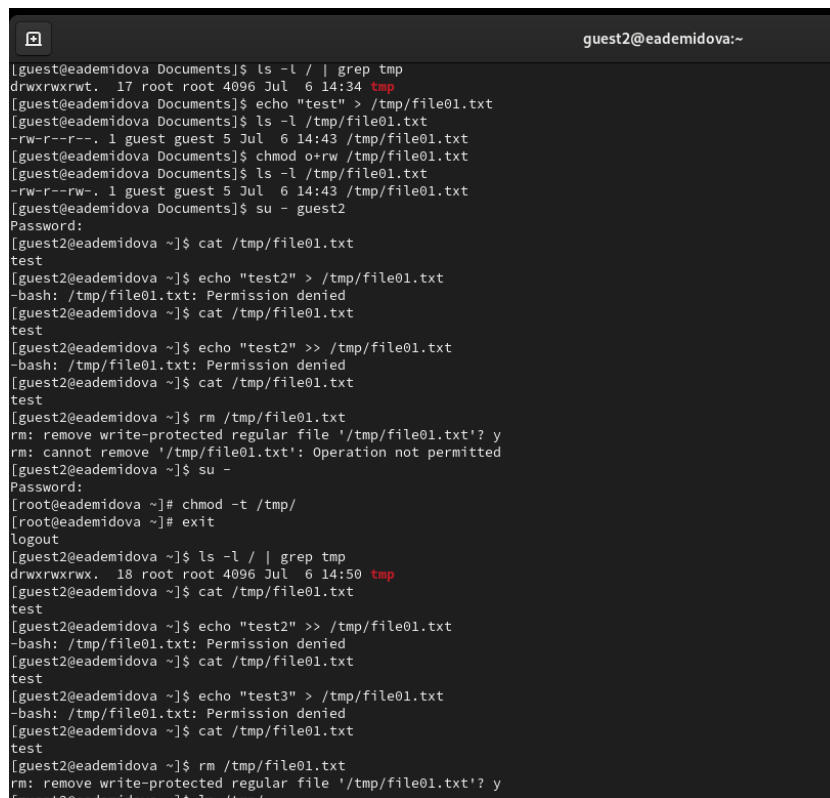
Сменим у программы readfile владельца и установим SetUID-бит. Теперь эта программа может прочитать файл readfile.c даже с пользователя guest, также она может прочитать файл /etc/shadow, владельцем которого guest также не является, так как программа readfile теперь имеет все права пользователя root(рис. 3.10):

```
[root@eademidova Documents]# chown root:guest /home/guest/Documents/readfile
[root@eademidova Documents]# chmod u+s /home/guest/Documents/readfile
[root@eademidova Documents]# exit
logout
[guest@eademidova Documents]$ ./readfile readfile.c
#include <fcntl.h>
[guest@eademidova Documents]$ ./readfile /etc/sh
shadow shadow- shells
[guest@eademidova Documents]$ ./readfile /etc/shadow
root:$6$c/APB9DR[guest@eademidova Documents]$
```

Рис. 3.10: Установка SetUID-бита на исполняемый файл readfile и проверка прав

После завершения установки операционной системы корректно перезапустим виртуальную машину и при запросе примем условия лицензии.

Проверим, что установлен атрибут Sticky на директории /tmp(в конце стоит t). Затем от имени пользователя guest создадим файл file01.txt в директории /tmp со словом test, затем просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные». После этого от пользователя guest2 попробуем дозаписать в этот файл новое слово, однако получим отказ, также нам отказано в перезаписи и удалении этого файла. Если же убрать Sticky бит, то нам будет разрешено удаление этого файла(рис. 3.11):



```
guest2@eademidova:~$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Jul  6 14:34 tmp
guest2@eademidova:~$ echo "test" > /tmp/file01.txt
guest2@eademidova:~$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Jul  6 14:43 /tmp/file01.txt
guest2@eademidova:~$ chmod o+rw /tmp/file01.txt
guest2@eademidova:~$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Jul  6 14:43 /tmp/file01.txt
guest2@eademidova:~$ su - guest2
Password:
guest2@eademidova:~$ cat /tmp/file01.txt
test
guest2@eademidova:~$ echo "test2" > /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
guest2@eademidova:~$ cat /tmp/file01.txt
test
guest2@eademidova:~$ echo "test2" >> /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
guest2@eademidova:~$ cat /tmp/file01.txt
test
guest2@eademidova:~$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
guest2@eademidova:~$ su -
Password:
root@eademidova:~$ chmod -t /tmp/
root@eademidova:~$ exit
logout
guest2@eademidova:~$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 Jul  6 14:50 tmp
guest2@eademidova:~$ cat /tmp/file01.txt
test
guest2@eademidova:~$ echo "test2" >> /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
guest2@eademidova:~$ cat /tmp/file01.txt
test
guest2@eademidova:~$ echo "test3" > /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
guest2@eademidova:~$ cat /tmp/file01.txt
test
guest2@eademidova:~$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
guest2@eademidova:~$ ls -l /tmp/
```

Рис. 3.11: Подключение образа диска дополнений

## 4 Выводы

В результате выполнения работы были выполнены:

- 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.

# Список литературы

1. Граннеман С. Скотт Граннеман: Linux. Карманный справочник. 2-е изд. Вильямс, 2019. 464 с.