

Лабораторная работа №5

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

Аскеров Александр Эдуардович

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Создание программы	5
2.2	Исследование Sticky-бита	12
3	Выводы	16
	Список литературы	17

Список иллюстраций

2.1	Отключение системы запретов до перезагрузки	5
2.2	Смена пользователя на guest	5
2.3	Создание программы simpleid.c	5
2.4	Программа simpleid.c	6
2.5	Компиляция программы	6
2.6	Отработка программы	6
2.7	Команда id	6
2.8	Программа simpleid2.c	7
2.9	Компиляция и отработка программы	7
2.10	Выполнение команд chown и chmod	7
2.11	Проверка	8
2.12	Запуск simpleid2 и id	8
2.13	Повтор действий относительно SetGID-бита	8
2.14	Программа readfile.c	9
2.15	Компиляция программы	9
2.16	Смена владельца и изменение прав	10
2.17	Проверка	10
2.18	Смена владельца readfile и установка SetU'D-бита	10
2.19	Проверка	11
2.20	Проверка	11
2.21	Проверка	12
2.22	Создание файла file01.txt	12
2.23	Изменение прав на file01.txt	12
2.24	Чтение файла file01.txt	12
2.25	Изменение файла file01.txt	13
2.26	Проверка	13
2.27	Попытка дозаписи в файл file01.txt	13
2.28	Проверка	13
2.29	Попытка удалить файл file01.txt	13
2.30	Снятие атрибута t с директории /tmp	14
2.31	Смена пользователя	14
2.32	Проверка	14
2.33	Повтор действий	15
2.34	Возврат атрибута t директории /tmp	15

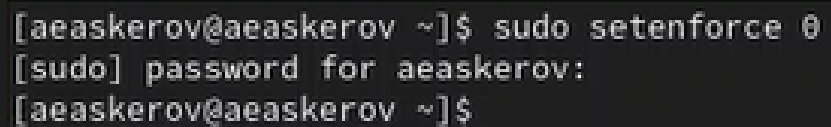
1 Цель работы

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 Выполнение лабораторной работы

2.1 Создание программы

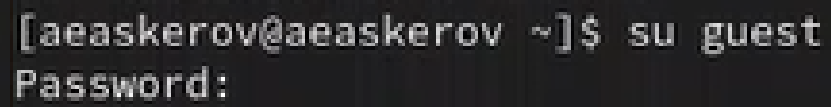
Отключим систему запретов до очередной перезагрузки системы.



```
[aeaskerov@aeaskerov ~]$ sudo setenforce 0
[sudo] password for aeaskerov:
[aeaskerov@aeaskerov ~]$
```

Рис. 2.1: Отключение системы запретов до перезагрузки

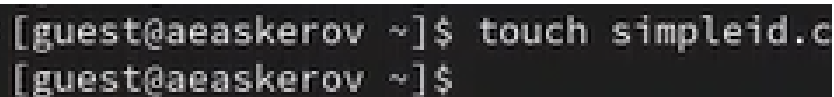
Войдём в систему от имени пользователя guest.



```
[aeaskerov@aeaskerov ~]$ su guest
Password:
```

Рис. 2.2: Смена пользователя на guest

Создадим программу simpleid.c.



```
[guest@aeaskerov ~]$ touch simpleid.c
[guest@aeaskerov ~]$
```

Рис. 2.3: Создание программы 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 = geteuid ();
9     gid_t gid = getegid ();
10    printf ("uid=%d, gid=%d\n", uid, gid);
11    return 0;
12 }

```

Рис. 2.4: Программа simpleid.c

Скомпилируем программу и убедимся, что файл программы создан.

```

[guest@aeaskerov ~]$ gcc simpleid.c -o simpleid
[guest@aeaskerov ~]$ ls
dir1 Documents Pictures simpleid simpleid.c
[guest@aeaskerov ~]$

```

Рис. 2.5: Компиляция программы

Выполним программу simpleid.

```

[guest@aeaskerov ~]$ ./simpleid
uid=1001, gid=100
[guest@aeaskerov ~]$

```

Рис. 2.6: Отработка программы

Выполним системную программу id.

```

[guest@aeaskerov ~]$ id
uid=1001(guest) gid=100(users) groups=100(users) context=unconfined_u:unconfined
_r:unconfined_t:s0-s0:c0.c1023
[guest@aeaskerov ~]$

```

Рис. 2.7: Команда id

Видим, что выводы программы и команды id совпадают.

Усложним программу, добавив вывод действительных идентификаторов.

```
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
17     return 0;
18 }
```

Рис. 2.8: Программа simpleid2.c

Скомпилируем и запустим simpleid2.c.

```
[guest@aeaskerov ~]$ gcc simpleid2.c -o simpleid2
[guest@aeaskerov ~]$ ./simpleid2
e_uid=1001, e_gid=100
real_uid=1001, real_gid=100
[guest@aeaskerov ~]$
```

Рис. 2.9: Компиляция и отработка программы

От имени суперпользователя выполним следующие команды.

```
[guest@aeaskerov ~]$ su -
Password:
[root@aeaskerov ~]# chown root:guest /home/guest/simpleid2
[root@aeaskerov ~]# chmod u+s /home/guest/simpleid2
[root@aeaskerov ~]#
```

Рис. 2.10: Выполнение команд chown и chmod

Теперь пользователь root будет владельцем файла, а группа guest будет группой этого файла. Вторая команда устанавливает бит установки SUID для файла simpleid2, позволяющий запускать файл с привилегиями пользователя-владельца файла, а не пользователя, который запускает его.

Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2.

```
[guest@aeaskerov ~]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 24488 Apr 11 18:18 simpleid2
[guest@aeaskerov ~]$
```

Рис. 2.11: Проверка

Запустим simpleid2 и id.

```
[guest@aeaskerov ~]$ ./simpleid2
e_uid=0, e_gid=100
real_uid=1001, real_gid=100
[guest@aeaskerov ~]$ id
uid=1001(guest) gid=100(users) groups=100(users) context=unconfined_u:unconfined
_r:unconfined_t:s0-s0:c0.c1023
[guest@aeaskerov ~]$
```

Рис. 2.12: Запуск simpleid2 и id

Результаты отличаются.

Проделаем тоже самое относительно SetGID-бита.

```
[guest@aeaskerov ~]$ su -
Password:
[root@aeaskerov ~]# chown root:guest /home/guest/simpleid2
[root@aeaskerov ~]# chmod g+s /home/guest/simpleid2
[root@aeaskerov ~]# exit
logout
[guest@aeaskerov ~]$ ls -l simpleid2
-rwxr-sr-x. 1 root guest 24488 Apr 11 18:18 simpleid2
[guest@aeaskerov ~]$ ./simpleid2
e_uid=1001, e_gid=1006
real_uid=1001, real_gid=100
[guest@aeaskerov ~]$ id
uid=1001(guest) gid=100(users) groups=100(users) context=unconfined_u:unconfined
_r:unconfined_t:s0-s0:c0.c1023
[guest@aeaskerov ~]$
```

Рис. 2.13: Повтор действий относительно SetGID-бита

Создадим программу readfile.c.


```

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
8 main (int argc, char* argv[])
9 {
10     unsigned char buffer[16];
11     size_t bytes_read;
12     int i;
13
14     int fd = open (argv[1], O_RDONLY);
15     do
16     {
17         bytes_read = read (fd, buffer, sizeof (buffer));
18         for (i = 0; i < bytes_read; ++i) printf("%c", buffer[i]);
19     }
20     while (bytes_read == sizeof (buffer));
21     close (fd);
22
23     return 0;
24 }

```

Рис. 2.14: Программа readfile.c

Откомпилируем её.

```

[guest@aeaskerov ~]$ gcc readfile.c -o readfile
[guest@aeaskerov ~]$

```

Рис. 2.15: Компиляция программы

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

```
[guest@aeaskerov ~]$ su -
Password:
[root@aeaskerov ~]# cd /home/guest
[root@aeaskerov guest]# chown root readfile.c
[root@aeaskerov guest]# ls -l
total 80
drwxrwxrwx. 2 guest users    19 Mar 14 12:49 dir
drwxr-xr-x. 2 guest users    6 Sep 12 2023 Documents
drwxr-xr-x. 2 guest users    6 Sep 12 2023 Pictures
-rwxr-xr-x. 1 guest users 24432 Apr 11 18:30 readfile
-rw-r--r--. 1 root users 470 Apr 11 18:30 readfile.c
-rwxr-xr-x. 1 guest users 24384 Apr 11 18:13 simpleid
-rwxr-sr-x. 1 root guest 24488 Apr 11 18:18 simpleid2
-rw-r--r--. 1 guest users 346 Apr 11 18:17 simpleid2.c
[root@aeaskerov guest]# chmod 000 readfile.c
[root@aeaskerov guest]#
```

Рис. 2.16: Смена владельца и изменение прав

Проверим, что пользователь guest не может прочитать файл readfile.c.

```
[guest@aeaskerov ~]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@aeaskerov ~]$
```

Рис. 2.17: Проверка

Сменим у программы readfile владельца и установим SetU'D-бит.

```
[guest@aeaskerov ~]$ su -
Password:
[root@aeaskerov ~]# cd /home/guest
[root@aeaskerov guest]# chown root readfile
[root@aeaskerov guest]# ls -l
total 80
drwxrwxrwx. 2 guest users    19 Mar 14 12:49 dir
drwxr-xr-x. 2 guest users    6 Sep 12 2023 Documents
drwxr-xr-x. 2 guest users    6 Sep 12 2023 Pictures
-rwxr-xr-x. 1 root users 24432 Apr 11 18:30 readfile
-----. 1 root users 470 Apr 11 18:30 readfile.c
-rwxr-xr-x. 1 guest users 24384 Apr 11 18:13 simpleid
-rwxr-sr-x. 1 root guest 24488 Apr 11 18:18 simpleid2
-rw-r--r--. 1 guest users 346 Apr 11 18:17 simpleid2.c
[root@aeaskerov guest]# chmod u+s readfile
[root@aeaskerov guest]#
```

Рис. 2.18: Смена владельца readfile и установка SetU'D-бита

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

```

[root@aeaskerov guest]# su - guest
[guest@aeaskerov ~]$ ./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@aeaskerov ~]$

```

Рис. 2.19: Проверка

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

```

avahi:!!:19606:~::~:
rtkit:!!:19606:~::~:
sssd:!!:19606:~::~:
pipewire:!!:19606:~::~:
libstoragemgmt:!*:19606:~::~:
systemd-oom:!*:19606:~::~:
tss:!!:19606:~::~:
geoclue:!!:19606:~::~:
cockpit-ws:!!:19606:~::~:

```

Рис. 2.20: Проверка

2.2 Исследование Sticky-бита

Выясним, установлен ли атрибут Sticky на директории /tmp.

```
[guest@aeaskerov ~]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 Apr 11 18:38 tmp
[guest@aeaskerov ~]$
```

Рис. 2.21: Проверка

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

```
[guest@aeaskerov ~]$ echo "test" > /tmp/file01.txt
[guest@aeaskerov ~]$
```

Рис. 2.22: Создание файла file01.txt

Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные».

```
[guest@aeaskerov ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest users 5 Apr 11 18:39 /tmp/file01.txt
[guest@aeaskerov ~]$ chmod o+rw /tmp/file01.txt
[guest@aeaskerov ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest users 5 Apr 11 18:39 /tmp/file01.txt
[guest@aeaskerov ~]$
```

Рис. 2.23: Изменение прав на file01.txt

От пользователя guest2 (не являющегося владельцем) попробуем прочитать файл /tmp/file01.txt.

```
[guest@aeaskerov ~]$ su - guest2
Password:
[guest2@aeaskerov ~]$ cat /tmp/file01.txt
test
[guest2@aeaskerov ~]$
```

Рис. 2.24: Чтение файла file01.txt

От пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2.

```
[guest2@aeaskerov ~]$ echo "test2" > /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
[guest2@aeaskerov ~]$
```

Рис. 2.25: Изменение файла file01.txt

Проверим содержимое файла.

```
[guest2@aeaskerov ~]$ cat /tmp/file01.txt
test
[guest2@aeaskerov ~]$
```

Рис. 2.26: Проверка

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

```
[guest2@aeaskerov ~]$ echo "test3" > /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
[guest2@aeaskerov ~]$
```

Рис. 2.27: Попытка дозаписи в файл file01.txt

Доступ запрещён.

Проверим содержимое файла.

```
[guest2@aeaskerov ~]$ cat /tmp/file01.txt
test
[guest2@aeaskerov ~]$
```

Рис. 2.28: Проверка

От пользователя guest2 попробуем удалить файл /tmp/file01.txt.

```
[guest2@aeaskerov ~]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@aeaskerov ~]$
```

Рис. 2.29: Попытка удалить файл file01.txt

Удалить файл не удалось.

Повысим свои права до суперпользователя и выполним после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp.

```
[guest2@aeaskerov ~]$ su -  
Password:  
[root@aeaskerov ~]# chmod -t /tmp  
[root@aeaskerov ~]#
```

Рис. 2.30: Снятие атрибута t с директории /tmp

Покинем режим суперпользователя.

```
[root@aeaskerov ~]# exit  
logout  
[guest2@aeaskerov ~]$
```

Рис. 2.31: Смена пользователя

От пользователя guest2 проверим, что атрибута t у директории /tmp нет.

```
[guest2@aeaskerov ~]$ ls -l / | grep tmp  
drwxrwxrwx. 15 root root 4096 Apr 11 18:44 tmp  
[guest2@aeaskerov ~]$
```

Рис. 2.32: Проверка

Повторим предыдущие шаги.

```
[guest2@aeaskerov ~]$ ls -l / | grep tmp
drwxrwxrwx. 15 root root 4096 Apr 11 18:44 tmp
[guest2@aeaskerov ~]$ echo "test2" >> /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
[guest2@aeaskerov ~]$ cat /tmp/file01.txt
test
[guest2@aeaskerov ~]$ echo "test3" > /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
[guest2@aeaskerov ~]$ cat /tmp/file01.txt
test
[guest2@aeaskerov ~]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
[guest2@aeaskerov ~]$
```

Рис. 2.33: Повтор действий

Ничего не изменилось, за исключением того, что файл получилось удалить от имени пользователя, не являющегося его владельцем.

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

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

Рис. 2.34: Возврат атрибута `t` директории `/tmp`

3 Выводы

Изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами. Рассмотрена работа механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Список литературы

1. How does the sticky bit work