

Лабораторная работа №5

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

Тазаева Анастасия Анатольевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Создание программы	6
2.2	Исследование Sticky-бита	11
3	Выводы	13

Список иллюстраций

2.1	simpleid.c	6
2.2	Компиляция файла simpleid.c	6
2.3	Выполнение программы simpleid	7
2.4	Системная программа id	7
2.5	simpleid2	7
2.6	Компиляция и выполнение программы simpleid2	7
2.7	Выполнение команд	8
2.8	Проверка правильности установки новых атрибутов	8
2.9	Выполнение программы simpleid2 и id	8
2.10	readfile.c	9
2.11	Компиляция readfile.c	9
2.12	Изменение владельца и права у файла readfile.c	9
2.13	Попытка прочитать файл readfile.c	10
2.14	Изменение владельца и права у программы readfile	10
2.15	Чтение файла readfile.c	10
2.16	Чтение файла /etc/shadow	11
2.17	Снятие атрибута t	12
2.18	Работа с файлом без атрибута t	12
2.19	Возвращение атрибута t	12

Список таблиц

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

1. Вошла в систему от имени пользователя guest.
2. Создала программу simpleid.c (рис. 1).

```
[guest@aatazaeva ~]$ ls
dir1      Видео      Загрузки      Музыка      'Рабочий стол'
simpleid.c Документы  Изображения  Общедоступные  Шаблоны
[guest@aatazaeva ~]$ nano simpleid.c
[guest@aatazaeva ~]$ cat 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;
}
```

Рис. 2.1: simpleid.c

3. Скомпилировала программу (рис. 2).

```
[guest@aatazaeva ~]$ gcc simpleid.c -o simpleid
[guest@aatazaeva ~]$ ls
dir1      simpleid.c  Документы  Изображения  Общедоступные  Шаблоны
simpleid   Видео      Загрузки      Музыка      'Рабочий стол'
```

Рис. 2.2: Компиляция файла simpleid.c

4. Выполнила программу (рис. 3).

```
[guest@aatazaeva ~]$ ./simpleid
uid=1001, gid=1001
```

Рис. 2.3: Выполнение программы simpleid

5. Выполнила системную программу id (рис. 4), значения выполнения программы simpleid и системной программы id совпадают.

```
[guest@aatazaeva ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 2.4: Системная программа id

6. Перезаписала прошлую программу, усложнив её (рис. 5).

```
[guest@aatazaeva ~]$ nano simpleid.c
[guest@aatazaeva ~]$ ls
dir1  simpleid2.c  Видео  Загрузки  Музыка  'Рабочий стол'
simpleid  simpleid.c  Документы  Изображения  Общедоступные  Шаблоны
[guest@aatazaeva ~]$ cat 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;
}
```

Рис. 2.5: simpleid2

7. Скомпилировала и запустила simpleid2.c (рис. 6).

```
[guest@aatazaeva ~]$ gcc simpleid2.c -o simpleid2
[guest@aatazaeva ~]$ ls
dir1  simpleid2  simpleid.c  Документы  Изображения  Общедоступные  Шаблоны
simpleid  simpleid2.c  Видео  Загрузки  Музыка  'Рабочий стол'
[guest@aatazaeva ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Рис. 2.6: Компиляция и выполнение программы simpleid2

8. От имени суперпользователя выполнила команды (рис. 7).

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

Рис. 2.7: Выполнение команд

9. Команда **chown root:guest simpleid2** устанавливает root-а владельцем файла, а также устанавливает группу guest владельцем этого же файла. Второй командой изменили права.
10. Выполнила проверку правильности установки новых атрибутов и смены владельца файла (рис. 8).

```
[guest@aatazaeva ~]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 17656 ноя 10 11:59 simpleid2
```

Рис. 2.8: Проверка правильности установки новых атрибутов

11. Запустила simpleid2 и id, результаты свпадают (рис. 9).

```
[guest@aatazaeva ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aatazaeva ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 2.9: Выполнение программы simpleid2 и id

12. Прodelала пункты 8-11 относительно SetGID-бита, сначала изменив права (рис. 10), далее выполнив программу и сверив с id (рис. 11).

```
[guest@aatazaeva ~]$ ls -l simpleid2
-rwsr-sr-x. 1 root guest 17656 ноя 10 11:59 simpleid2
[guest@aatazaeva ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aatazaeva ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@aatazaeva ~]# chmod g+s /home/guest/simpleid2
```

13. Создала программу readfile.c (рис. 12).


```
[guest@aatazaeva ~]$ touch readfile.c
[guest@aatazaeva ~]$ nano readfile.c
[guest@aatazaeva ~]$ cat 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;
}
```

Рис. 2.10: readfile.c

14. Скомпилировала её (рис. 13).

```
[guest@aatazaeva ~]$ gcc readfile.c -o readfile
[guest@aatazaeva ~]$ ls
dir1      simpleid      simpleid.c    Загрузки      Общедоступные
readfile  simpleid2     Видео         Изображения   'Рабочий стол'
readfile.c simpleid2.c   Документы     Музыка         Шаблоны
[guest@aatazaeva ~]$ ls -l
итого 72
drwx--x---. 2 guest guest    19 ноя 10 10:39 dir1
-rwxr-xr-x. 1 guest guest 17600 ноя 10 12:27 readfile
-rw-r--r--. 1 guest guest   421 ноя 10 12:25 readfile.c
```

Рис. 2.11: Компиляция readfile.c

15. Сменила владельца и изменила права у файла readfile.c (рис. 15).

```
[root@aatazaeva ~]# chown root:guest /home/guest/readfile.c
[root@aatazaeva ~]# chmod g-r /home/guest/readfile.c
```

Рис. 2.12: Изменение владельца и права у файла readfile.c

16. Пользователь guest не может прочитать файл readfile.c (рис. 15).

```
[guest@aatazaeva ~]$ ls -l readfile.c
-rw----r--. 1 root guest 421 ноя 10 12:25 readfile.c
[guest@aatazaeva ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
```

Рис. 2.13: Попытка прочитать файл readfile.c

17. Сменила владельца и изменила права у программы readfile (рис. 17).

```
[root@aatazaeva ~]# chown root:guest /home/guest/readfile
[root@aatazaeva ~]# chmod u+s /home/guest/readfile
```

Рис. 2.14: Изменение владельца и права у программы readfile

18. Чтение файла readfile.c с помощью программы readfile (рис. 17).

```
[guest@aatazaeva ~]$ ./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;
}
```

Рис. 2.15: Чтение файла readfile.c

19. Чтение файла /etc/shadow с помощью программы readfile (рис. 18).

имеющегося (не удалось)(рис. 23), удалить файл- также не удалось(рис. 24).

```
[guest@aatazaeva ~]$ echo "test" > /tmp/file01.txt
[guest2@aatazaeva ~]$ echo "test2"
-bash: /tmp/file01.txt: Отказано в
[guest2@aatazaeva ~]$ echo "test3" >> /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@aatazaeva ~]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл
rm: невозможно удалить '/tmp/file01.txt': Опер
```

3. Повысили права пользователя и сняли атрибут t(рис. 25).

```
[guest2@aatazaeva ~]$ su -
Пароль:
[root@aatazaeva ~]# chmod -t /tmp/
[root@aatazaeva ~]# exit
ВЫХОД
```

Рис. 2.17: Снятие атрибута t

4. Проверили атрибут у директории /tmp и проверили шаг 2(рис. 26), удалось только удалить файл.

```
[guest2@aatazaeva ~]$ ls -l / | grep tmp
drwxrwxrwx. 25 root root 4096 ноя 10 12:43 tmp
[guest2@aatazaeva ~]$ cat /tmp/file01.txt
test
[guest2@aatazaeva ~]$ echo "test2_t" > /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@aatazaeva ~]$ echo "test3_t" >> /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@aatazaeva ~]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? yes
```

Рис. 2.18: Работа с файлом без атрибута t

5. Вернули атрибут t(рис. 27).

```
[root@aatazaeva ~]# chmod +t /tmp/
[root@aatazaeva ~]# exit
ВЫХОД
[guest2@aatazaeva ~]$ ls -l / | grep tmp
drwxrwxrwt. 27 root root 4096 ноя 10 12:46 tmp
```

Рис. 2.19: Возвращение атрибута t

3 Выводы

Изучила механизм изменения идентификаторов, применения SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.