Лабораторная работа №5

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

Федотов Дмитрий Константинович

Содержание

1 Цель работы **5**

2 Задание **6**

3 Выполнение лабораторной работы **7**

4 Выводы **16**

Список иллюстраций

3.1 Установка компилятора gcc 7
3.2 Отключение системы запретов 7
3.3 Проверка названий компиляторов 8
3.4 Создание файла simpleid.c 8
3.5 Создание программы simpleid.c 8
3.6 Компиляция программы 8
3.7 Выполнение созданной программы 9
3.8 Усложнение программы 9
3.9 Компиляция и запуск файла 9
3.10 Смена владельца и атрибутов от имени суперпользователя 10
3.11 Проверка id пользователя и группы 10
3.12 Создание программы `readfile.c` 10
3.13 Компиляция программы 10
3.14 Смена владельца и изменение прав файла 11
3.15 Попытка прочесть файл 11
3.16 Смена владельца и установка SetUID-бита 11
3.17 Проверка чтения файла 12
3.18 Проверка чтения файла `/etc/shadow` 12
3.19 Проверка нахождения атрибута Sticky на директории `/tmp` 13
3.20 Создание файла и внесение записи в него 13
3.21 Просмотр атрибутов файла и установление прав на чтение и запись для категории “все остальные” 13
3.22 Чтение файла от имени пользователя `guest2` 13
3.23 Дозапись слова в файл от имени пользователя `guest2` 14
3.24 Попытка удаления файла от имени пользователя `guest2` 14
3.25 Повышение прав до суперпользователя. Снятие атрибута `t` 14
3.26 Повтор предыдущих шагов 15
3.27 Переход в режим суперпользователя и возврат атрибута `t` 15

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. [1]

2 Задание

1. Подготовить лабораторный стенд
2. Рассмотреть компиляцию программ
3. Создать программы
4. Исследовать Sticky-бит

3 Выполнение лабораторной работы

1. Установил компилятор gcc с помощью команды `yum install gcc` (рис -@fig:001).

Рис. 3.1: Установка компилятора gcc

Отключил систему защиты SELinux с помощью команды `setenforce 0`. После этого команда `getenforce` вывела `Permissive` (рис -@fig:002).

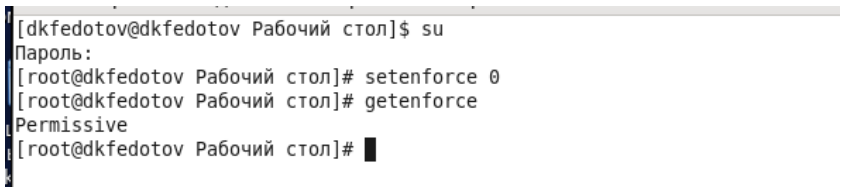

Рис. 3.2: Отключение системы запретов

2. Изучил компиляцию программ. Компилятор языка C называется gcc. Компилятор языка C++ называется g++ и запускается с параметрами почти так

же, как gcc. Проверил это с поомщью команд `whereis gcc` и `whereis g++` (рис -@fig:003).

```
[root@dkfedotov Рабочий стол]# whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/libexec/gcc /usr/share/man/man1/gcc.1.gz
[root@dkfedotov Рабочий стол]# where g++
bash: where: команда не найдена
[root@dkfedotov Рабочий стол]# whereis g++
g++:
[root@dkfedotov Рабочий стол]#
```

Рис. 3.3: Проверка названий компиляторов

3. Вошел в в систему от имени пользователя `guest` и создал программу `simpleid.c` (рис -@fig:004 и рис -@fig:005).

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

Рис. 3.4: Создание файла simpleid.c

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main ()
{
        uid_tuid = geteuid ();
        gid_tgid = getegid ();
        printf ("uid=%d, gid=%d\n", uid, gid);
        return 0;
}
```

Рис. 3.5: Создание программы simpleid.c

Скомпилировал программу (рис -@fig:006)

```
[guest@dkfedotov ~]$ gcc simpleid.c -o simpleid
[guest@dkfedotov ~]$
```

Рис. 3.6: Компиляция программы

Выполнил программу `simpleid` (рис -@fig:007)

```
[guest@dkfedotov ~]$ ./simpleid
uid=501, gid=501
```

Рис. 3.7: Выполнение созданной программы

Усложнил программу, добавив вывод действительных идентификаторов (рис -@fig:008)

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main ()
{
        uid_t real_uid = geteuid ();
        uid_t e_uid = geteuid ();

        gid_t real_gid = getegid ();
        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. 3.8: Усложнение программы

Скомпилировал и запустил `simpleid2.c` (рис -@fig:009)

```
[guest@dkfedotov ~]$ mv simpleid.c simpleid2.c
[guest@dkfedotov ~]$ gcc simpleid2.c -o simpleid2
[guest@dkfedotov ~]$ ./simpleid2
e_uid=501, e_gid=501
real_uid=501,           real_gid=501
[guest@dkfedotov ~]$
```

Рис. 3.9: Компиляция и запуск файла

От имени суперпользователя выполнил следующие команды (рис -@fig:010)

```
[root@dkfedotov guest]# chown root:guest /home/guest/simpleid2
[root@dkfedotov guest]# chmod u+s /home/guest/simpleid2
[root@dkfedotov guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 4971 Ноя 13 18:50 simpleid2
[root@dkfedotov guest]#
```

Рис. 3.10: Смена владельца и атрибутов от имени суперпользователя

Запустил `simpleid2` и `id` (рис -@fig:011)

```
[root@dkfedotov guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0,             real_gid=0
[root@dkfedotov guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconf
ined_t:s0-s0:c0.c1023
[root@dkfedotov guest]#
```

Рис. 3.11: Проверка id пользователя и группы

Создал программу `readfile.c` (рис -@fig:012)

```
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
int
main (intargc,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. 3.12: Создание программы `readfile.c`

Откомпилировал созданную программу (рис -@fig:013)

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

Рис. 3.13: Компиляция программы

Сменил владельца у файла readfile.c и изменил права так, чтобы только суперпользователь мог прочитать его, а guest не мог (рис -@fig:015)

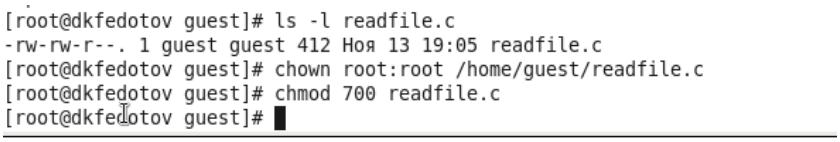

Рис. 3.14: Смена владельца и изменение прав файла

Проверил, что пользователь `guest` не может прочитать файл `readfile.c` (рис -@fig:016)

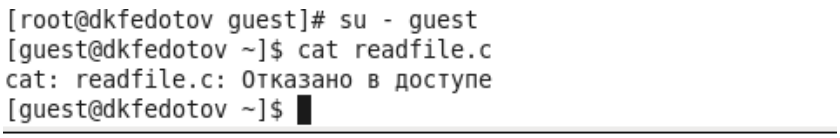

Рис. 3.15: Попытка прочесть файл

Сменил у программы `readfile` владельца и утсановил SetUID-бит (рис -@fig:017)

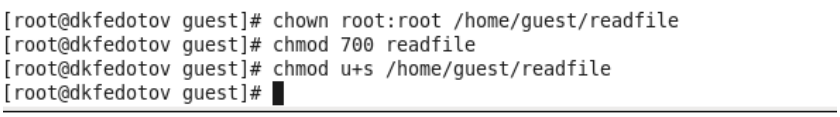

Рис. 3.16: Смена владельца и установка SetUID-бита

Проверил, может ли программа `readfile` прочитать файл `readfile.c`. Да, может. (рис -@fig:018)

```
Пароль:
[root@dkfedotov guest]# ./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;
}
[root@dkfedotov guest]#
```

Рис. 3.17: Проверка чтения файла

Проверил, может ли программа `readfile` прочитать файл `/etc/shadow`. Да, может. (рис -@fig:019)

```
}
[root@dkfedotov guest]# ./readfile /etc/shadow
root:$6$xwjSGg/2BQISngs7$8An9ZlNjmFCXXk9dmZTy3.SaCCTafmeLnu6.0cT5uV.2FQm630D9c
ciDRVB7k7yKdtMnmc11nz.N1qQ2clv0:18915:0:99999:7:::
bin:*:15980:0:99999:7:::
daemon:*:15980:0:99999:7:::
adm:*:15980:0:99999:7:::
lp:*:15980:0:99999:7:::
sync:*:15980:0:99999:7:::
shutdown:*:15980:0:99999:7:::
halt:*:15980:0:99999:7:::
mail:*:15980:0:99999:7:::
uucp:*:15980:0:99999:7:::
operator:*:15980:0:99999:7:::
games:*:15980:0:99999:7:::
gopher:*:15980:0:99999:7:::
```

Рис. 3.18: Проверка чтения файла `/etc/shadow`

4. Исследовал Sticky-бит

   Выяснил, что атрибут Sticky установлен на директорию `/tmp`, для чего выполнил команду `ls -l / | grep tmp` (рис -@fig:020)

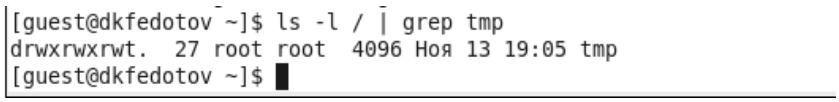

Рис. 3.19: Проверка нахождения атрибута Sticky на директории `/tmp`

От имени пользователя `guest` создал файл `file01.txt` в директории `/tmp` со словом `test` (рис -@fig:021):

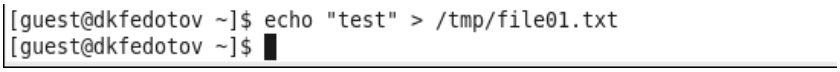

Рис. 3.20: Создание файла и внесение записи в него

Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей “все остальные” (рис -@fig:022):

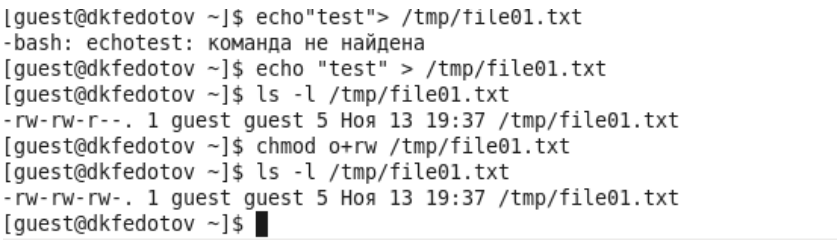

Рис. 3.21: Просмотр атрибутов файла и установление прав на чтение и запись для категории “все остальные”

От имени пользователя `guest2` (не являющегося владельцем) прочитал файл `/tmp/file01.txt` (рис -@fig:023):

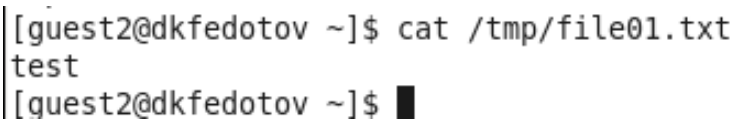

Рис. 3.22: Чтение файла от имени пользователя `guest2`

От имени пользователя `guest2` дозаписал в файл `/tmp/file01.txt` слово `test2` (рис -@fig:024):

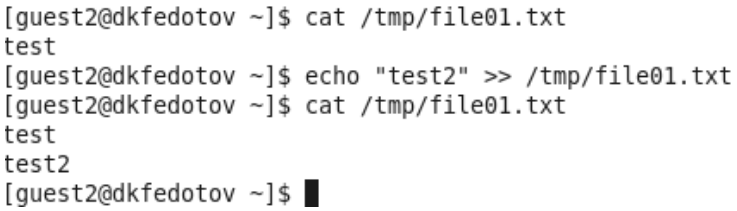

Рис. 3.23: Дозапись слова в файл от имени пользователя `guest2`

От имени пользователя `guest2` попробовал удалить файл `/tmp/file01.txt` (рис -@fig:025):

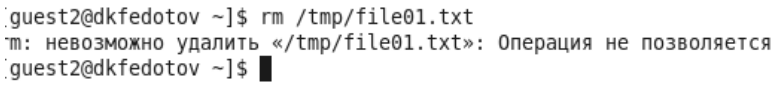

Рис. 3.24: Попытка удаления файла от имени пользователя `guest2`

Мне не удалось удалить файл.

Повысил свои права до суперпользователя и выполнил после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`(рис -@fig:026):

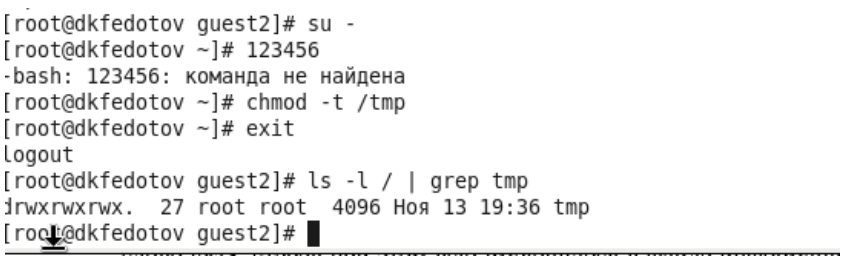

Рис. 3.25: Повышение прав до суперпользователя. Снятие атрибута `t`

Повторил предыдущие шаги (рис -@fig:029):

Рис. 3.26: Повтор предыдущих шагов

Как видно из рисунка, удалось выполнить все команды, которые были рассмотрены выше, включая удаление.

Повысил свои права до суперпользователя и вернул атрибут `t` на директорию `/tmp` (рис -@fig:030):

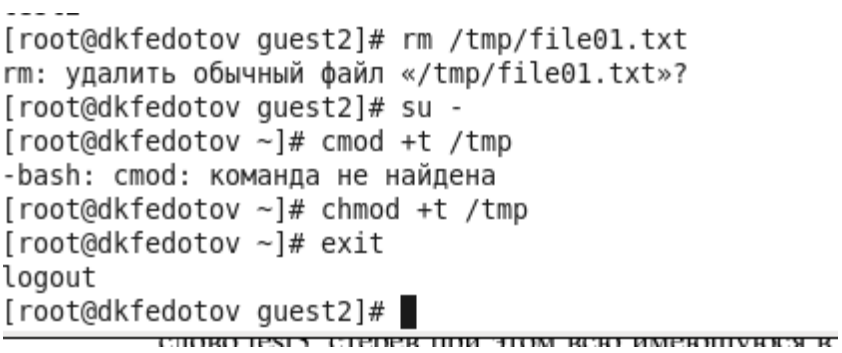

Рис. 3.27: Переход в режим суперпользователя и возврат атрибута `t`

4 Выводы

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