

Лабораторная работа №5

Что нужно было сделать?

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

Ход работы

- Вошел в систему от имени пользователя guest:

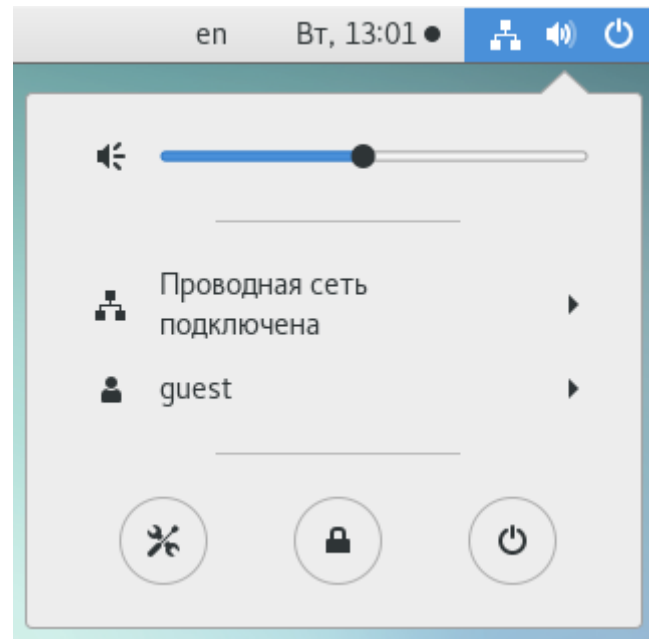


Рис. 1. Вход в систему

Ход работы

- Создал файл с кодом программы на языке C:

```
[guest@PodorogaVA ~]$ touch simpleid.c  
[guest@PodorogaVA ~]$ █
```

Рис. 2.1. Создание файла

```
#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", uid, gid);  
    return 0;  
}
```

Рис. 2.2. Код программы

Ход работы

- Скомпилировал программу:

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



Рис. 3. Компиляция

Ход работы

- Выполнил программу командой ./simpleid:

```
uid=1001, gid=1001[guest@PodorogaVA ~]$ █
```

Рис. 4. Результат работы

Ход работы

- Выполнил системную программу id:

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

Рис. 5. Выполнение id

Ход работы

- Усложнил программу, добавив вывод действительных идентификаторов:

```
[guest@PodorogaVA ~]$ touch simpleid2.c  
[guest@PodorogaVA ~]$ █
```

Рис. 6.1. Создание файла

```
#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. 6.2. Код программы

Ход работы

- Скомпилировал и запустил:

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

Рис. 7. Компиляция и запуск

Ход работы

- От имени суперпользователя выполнил следующие команды:

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

Рис. 8. Изменение атрибутов

Ход работы

- С помощью команды `chown root:guest /home/guest/simpleid2` удалось изменить права на файл с `guest` на `root`. Следующая команда позволяет установить атрибут `SetUID`, при наличии которого программа запускается с правами владельца файла.

Ход работы

- Выполнил проверку правильности установки прав и атрибутов:

```
[guest@PodorogaVA ~]$ ls -l simpleid2  
-rwsrwxr-x. 1 root guest 8616 окт  4 13:12 simpleid2  
[guest@PodorogaVA ~]$
```

*Рис. 10. Установка атрибута *i* на файл *file1**

Ход работы

- Запустил simpleid2 и id:

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

Рис. 11. Запуск simpleid2 и id

Ход работы

- Проделал то же самое относительно SetGID-бита.

Ход работы

- Создал программу readfile.c:

```
[root@PodorogaVA guest]# touch readfile.c  
[root@PodorogaVA guest]#
```

Рис. 13.1. Создание файла

```
#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. 13.2. Код программы

Ход работы

➤ Выполнил компиляцию:

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

Рис. 14. Компиляция

Ход работы

- Сменил владельца у файла readfile.c и изменил права так, чтобы только суперпользователь мог прочитать его, а guest не мог:

```
[root@PodorogaVA podorogava]# sudo chown root:guest /home/guest/readfile.c  
[root@PodorogaVA podorogava]# sudo chmod 700 /home/guest/readfile.c  
[root@PodorogaVA podorogava]#
```

Рис. 15. Смена прав и владельца

Ход работы

- Проверил, что пользователь guest не может прочитать файл readfile.c:

```
[guest@PodorogaVA ~]$ cat readfile.c  
cat: readfile.c: Отказано в доступе  
[guest@PodorogaVA ~]$
```

Рис. 16. Установка атрибута i на файл file1

При этом программа может прочитать файл readfile.c и /etc/shadow

Ход работы

- Выяснил, установлен ли атрибут Sticky на директории /tmp, далее от имени пользователя guest создал файл file01.txt со словом test, посмотрел атрибуты этого файла и разрешил чтение и запись для категории пользователей "все остальные", проверил правильность установки атрибутов:

```
[guest@PodorogaVA ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 окт  4 14:52 tmp
[guest@PodorogaVA ~]$ echo "test" > /tmp/file01.txt
[guest@PodorogaVA ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт  4 14:57 /tmp/file01.txt
[guest@PodorogaVA ~]$ chmod o+rx /tmp/file01.txt
[guest@PodorogaVA ~]$ ls -l /tmp/file01.txt
-rw-rw-r-x. 1 guest guest 5 окт  4 14:57 /tmp/file01.txt
[guest@PodorogaVA ~]$
```

Рис. 17. Исследование Sticky-бита

Ход работы

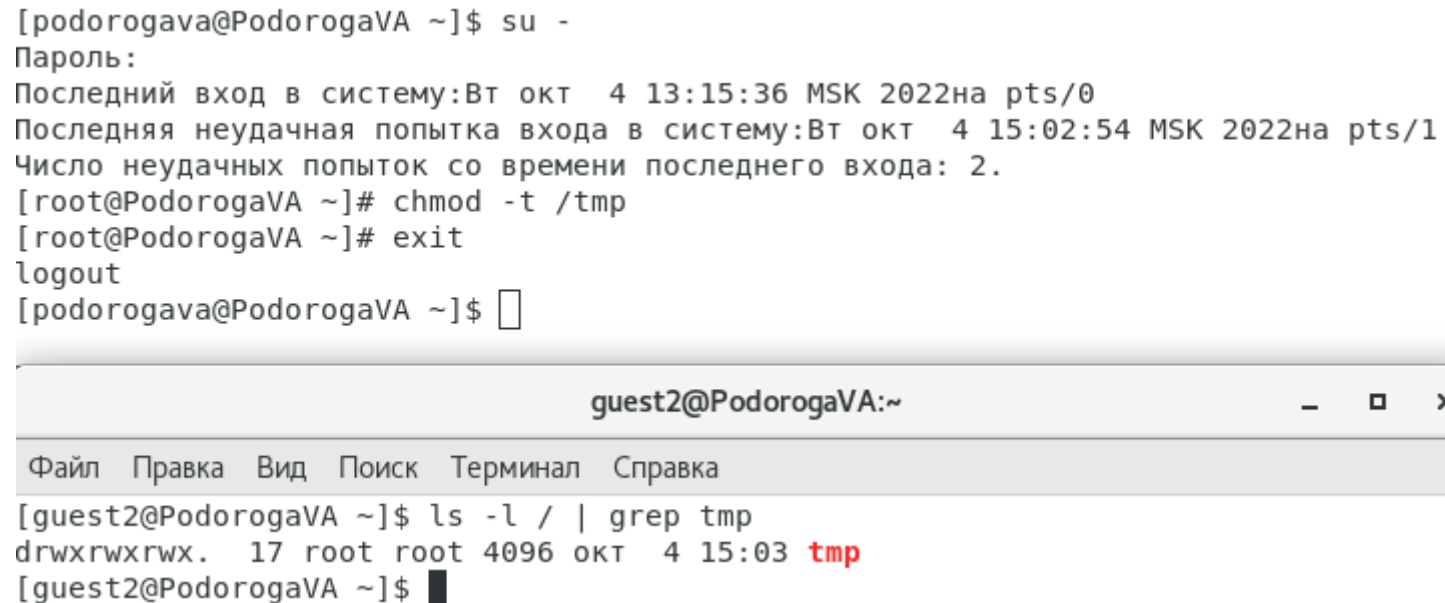
- От пользователя guest2 попробовал прочитать файл (удалось), попробовал дозаписать в файл test2 (удалось перезаписать), снова проверил содержимое и заменил содержимое файла на test3. Все эти операции выполнить удалось, а удалить - не получилось:

```
[podorogava@PodorogaVA ~]$ su - guest2
Пароль:
Последний вход в систему:Вт сен 20 18:35:53 MSK 2022на pts/2
[guest2@PodorogaVA ~]$ cat /tmp/file01.txt
test
[guest2@PodorogaVA ~]$ exho "test2" > /tmp/file01.txt
bash: exho: команда не найдена...
[guest2@PodorogaVA ~]$ echo "test2" > /tmp/file01.txt
[guest2@PodorogaVA ~]$ cat /tmp/file01.txt
test2
[guest2@PodorogaVA ~]$ exho "test3" > /tmp/file01.txt
bash: exho: команда не найдена...
[guest2@PodorogaVA ~]$ echo "test3" > /tmp/file01.txt
[guest2@PodorogaVA ~]$ cat /tmp/file01.txt
test3
[guest2@PodorogaVA ~]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
```

Рис. 18. Операции над файлом

Ход работы

- Повысил права до суперпользователя и снял Sticky-бит с использованием `chmod`, затем вышел из режима суперпользователя:



```
[podorogava@PodorogaVA ~]$ su -  
Пароль:  
Последний вход в систему:Вт окт  4 13:15:36 MSK 2022на pts/0  
Последняя неудачная попытка входа в систему:Вт окт  4 15:02:54 MSK 2022на pts/1  
Число неудачных попыток со времени последнего входа: 2.  
[root@PodorogaVA ~]# chmod -t /tmp  
[root@PodorogaVA ~]# exit  
logout  
[podorogava@PodorogaVA ~]$
```



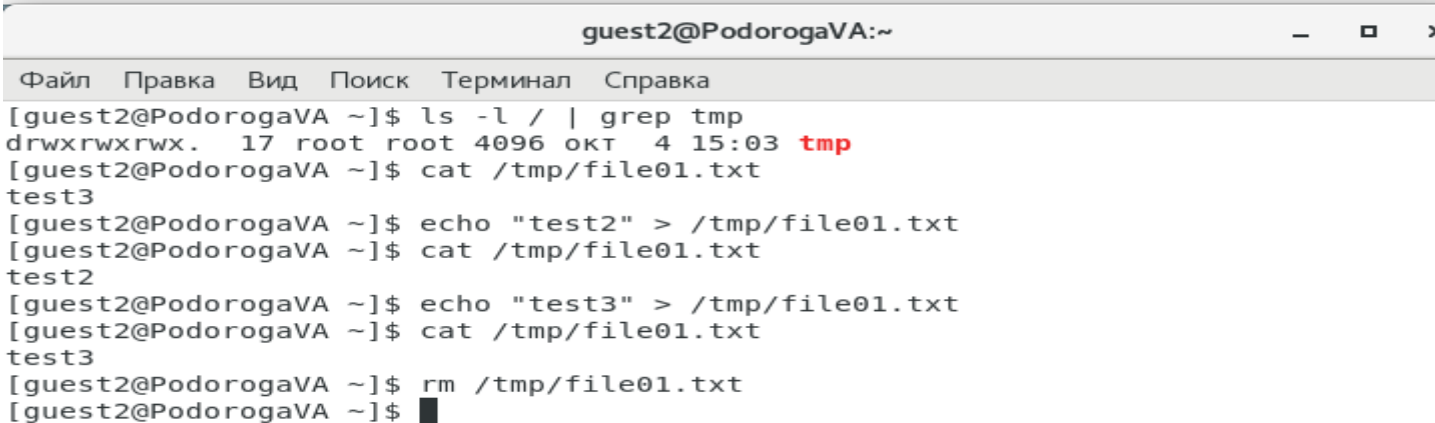
```
guest2@PodorogaVA:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
[guest2@PodorogaVA ~]$ ls -l / | grep tmp  
drwxrwxrwx. 17 root root 4096 окт  4 15:03 tmp  
[guest2@PodorogaVA ~]$
```

Рис. 19. Снятие Sticky-бита

Ход работы

- Попробовал снова выполнить вышеописанные операции с файлом file01.txt, на этот раз всё получилось:

```
[podorogava@PodorogaVA ~]$ su -
Пароль:
Последний вход в систему:Вт окт  4 13:15:36 MSK 2022на pts/0
Последняя неудачная попытка входа в систему:Вт окт  4 15:02:54 MSK 2022на pts/1
Число неудачных попыток со времени последнего входа: 2.
[root@PodorogaVA ~]# chmod -t /tmp
[root@PodorogaVA ~]# exit
logout
[podorogava@PodorogaVA ~]$
```



```
[guest2@PodorogaVA ~]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 окт  4 15:03 tmp
[guest2@PodorogaVA ~]$ cat /tmp/file01.txt
test3
[guest2@PodorogaVA ~]$ echo "test2" > /tmp/file01.txt
[guest2@PodorogaVA ~]$ cat /tmp/file01.txt
test2
[guest2@PodorogaVA ~]$ echo "test3" > /tmp/file01.txt
[guest2@PodorogaVA ~]$ cat /tmp/file01.txt
test3
[guest2@PodorogaVA ~]$ rm /tmp/file01.txt
[guest2@PodorogaVA ~]$
```

Рис. 20. Операции над файлом без атрибута *t* на директории *tmp*

Ход работы

- Повысил права до суперпользователя, вернул атрибут `t` на директорию:

```
[podorogava@PodorogaVA ~]$ su -  
Пароль:  
Последний вход в систему:Вт окт  4 15:03:11 MSK 2022на pts/1  
[root@PodorogaVA ~]# chmod +t /tmp  
[root@PodorogaVA ~]# exit  
logout  
[podorogava@PodorogaVA ~]$
```

Рис. 21. Возвращение атрибута `t` на директорию `tmp`

Вывод

В результате выполнения работы я изучил механизмы изменения идентификаторов, использования SetUID-бита и Sticky-бита, получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работу механизма смены идентификатора процессов пользователей, а также механизм влияния Sticky-бита на возможные операции над файлами внутри директории.

Спасибо за внимание!