

Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния
дополнительных атрибутов

Калинина Кристина Сергеевна

Содержание

Цель работы	5
Теоретические сведения	6
Выполнение лабораторной работы	8
Выводы	15
Список литературы	16

List of Figures

0.1	Создание программы simpleid.c	8
0.2	Код программы simpleid.c	8
0.3	Выполнение программы и сравнение результата с выводом команды id	9
0.4	Код программы simpleid2.c	9
0.5	Выполнение программы simpleid2.c	9
0.6	Смена владельца и установка атрибутов	10
0.7	Проверка и запуск программы	10
0.8	Установка атрибутов	10
0.9	Повтор действий с SetGID-битом	10
0.10	Код программы readfile.c	11
0.11	Компиляция программы readfile.c	11
0.12	Смена владельца файла readfile.c и прав на него	11
0.13	Отказ в чтении readfile.c	12
0.14	Смена владельца файла readfile и установка SetU'D-бит	12
0.15	Чтение readfile.c и “/etc/shadow”	12
0.16	Атрибут Sticky на директории /tmp	13
0.17	Создание file01.txt и смена атрибутов	13
0.18	Работа с file01.txt от guest2	13
0.19	Снятие атрибута Sticky с директории /tmp	14
0.20	Повторное выполнение команд от guest2	14
0.21	Возвращение атрибута Sticky на директорию /tmp	14

List of Tables

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.

Теоретические сведения

В операционных системах Linux используются 3 базовых права доступа – на чтение (read), запись (write) и исполнение (execute). Соответственно, права назначаются пользователю (user), группе (group) и всем остальным (world). [1]

Setuid – это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо. Классический пример использования этого бита в операционной системе это команда `sudo`. На месте, где обычно установлен классический бит `x` (на исполнение), выставлен специальный бит `s`. Это позволяет обычному пользователю системы выполнять команды с повышенными привилегиями без необходимости входа в систему как `root`, разумеется зная пароль пользователя `root`. Для установки используется команда “`chmod u+s`”. [1]

Принцип работы Setgid очень похож на setuid с отличием, что файл будет запускаться пользователем от имени группы, которая владеет файлом. Аналогично setuid, бит setgid выставляется с помощью команды `chmod g + s`. Удалить эти биты можно также командой `chmod`, только вместо «`+`» используется «`-`». [1]

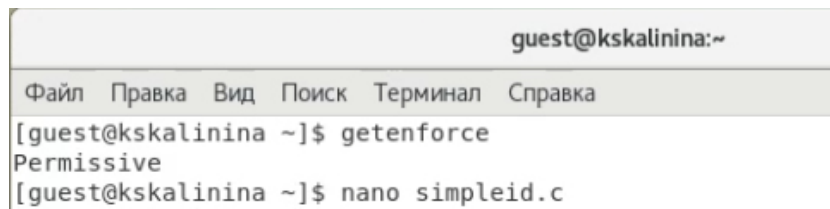
Третий из специальных разрешений — sticky bit. Это разрешение полезно для защиты файлов от случайного удаления в среде, где несколько пользователей имеют права на запись в один и тот же каталог. Если применяется закрепленный sticky bit, пользователь может удалить файл, только если он является пользователем-владельцем файла или каталога, в котором содержится файл. По этой причине он применяется в качестве разрешения по умолчанию для каталога `/tmp` и может быть

полезен также для каталогов общих групп. [2]

При использовании `ls -ld`, вы можете видеть sticky bit как `t` в позиции, где вы обычно видите разрешение на выполнение для других. Для sticky bit используйте `chmod +t`, а затем имя файла или каталога, для которого вы хотите установить разрешения. [2]

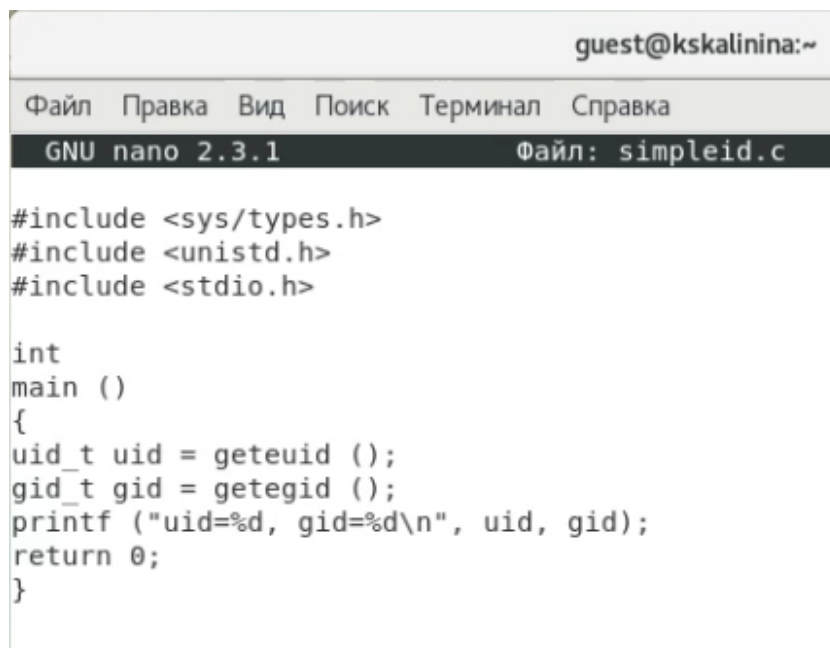
Выполнение лабораторной работы

1. Вошла в систему от имени пользователя guest. Создала программу simpleid.c (fig. 0.1, fig. 0.2).



```
guest@kskalinina:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@kskalinina ~]$ getenforce  
Permissive  
[guest@kskalinina ~]$ nano simpleid.c
```

Figure 0.1: Создание программы simpleid.c



```
guest@kskalinina:~  
Файл Правка Вид Поиск Терминал Справка  
GNU nano 2.3.1 Файл: 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;  
}
```

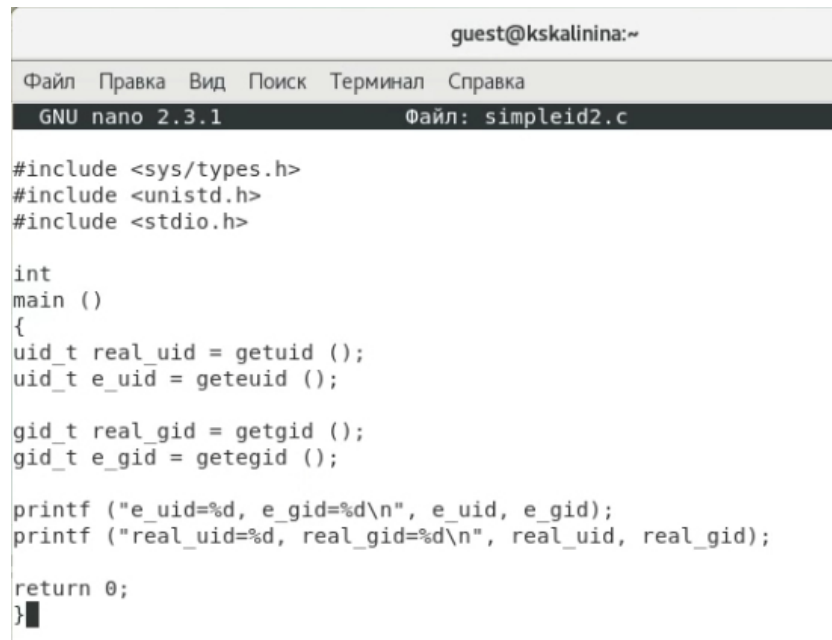
Figure 0.2: Код программы simpleid.c

2. Скомпилировала и выполнила программу. Выполнив системную программу `id` убедилась в правильности выведенных данных (fig. 0.3).

```
[guest@kskalinina ~]$ gcc simpleid.c -o simpleid
[guest@kskalinina ~]$ ./simpleid
uid=1001, gid=1001
[guest@kskalinina ~]$ id
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

Figure 0.3: Выполнение программы и сравнение результата с выводом команды `id`

3. Усложнила программу, добавив вывод действительных идентификаторов, назвала её `simpleid2.c` (fig. 0.4).



```
guest@kskalinina:~
Файл Правка Вид Поиск Терминал Справка
GNU nano 2.3.1 Файл: 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;
}
```

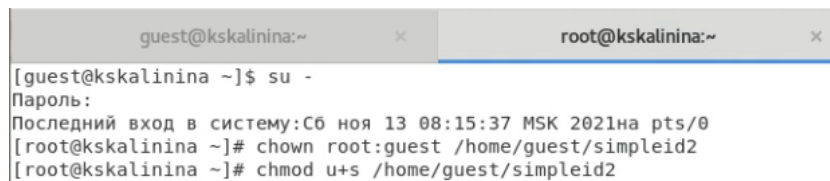
Figure 0.4: Код программы `simpleid2.c`

4. Скомпилировала и запустила программу (fig. 0.5).

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

Figure 0.5: Выполнение программы `simpleid2.c`

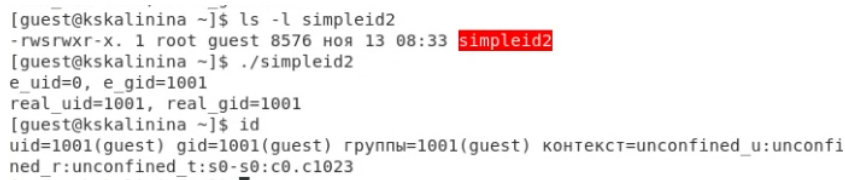
5. От имени суперпользователя установила новые атрибуты и сменила владельца файла simpleid2 (fig. 0.6).



```
guest@kskalinina:~ x root@kskalinina:~ x
[guest@kskalinina ~]$ su -
Пароль:
Последний вход в систему:Сб ноя 13 08:15:37 MSK 2021на pts/0
[root@kskalinina ~]# chown root:guest /home/guest/simpleid2
[root@kskalinina ~]# chmod u+s /home/guest/simpleid2
```

Figure 0.6: Смена владельца и установка атрибутов

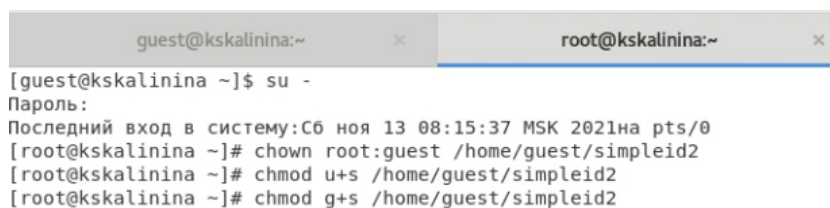
6. Выполнила проверку и запустила программу (fig. 0.7).



```
[guest@kskalinina ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 8576 ноя 13 08:33 simpleid2
[guest@kskalinina ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@kskalinina ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

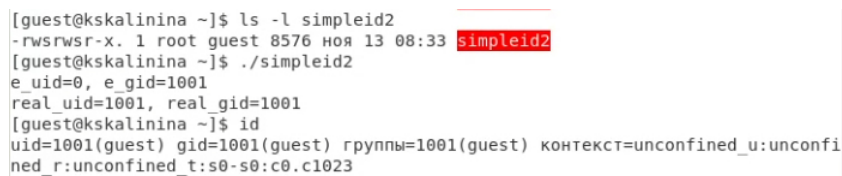
Figure 0.7: Проверка и запуск программы

7. Прodelала тоже самое относительно SetGID-бита (fig. 0.8, fig. 0.9).



```
guest@kskalinina:~ x root@kskalinina:~ x
[guest@kskalinina ~]$ su -
Пароль:
Последний вход в систему:Сб ноя 13 08:15:37 MSK 2021на pts/0
[root@kskalinina ~]# chown root:guest /home/guest/simpleid2
[root@kskalinina ~]# chmod u+s /home/guest/simpleid2
[root@kskalinina ~]# chmod g+s /home/guest/simpleid2
```

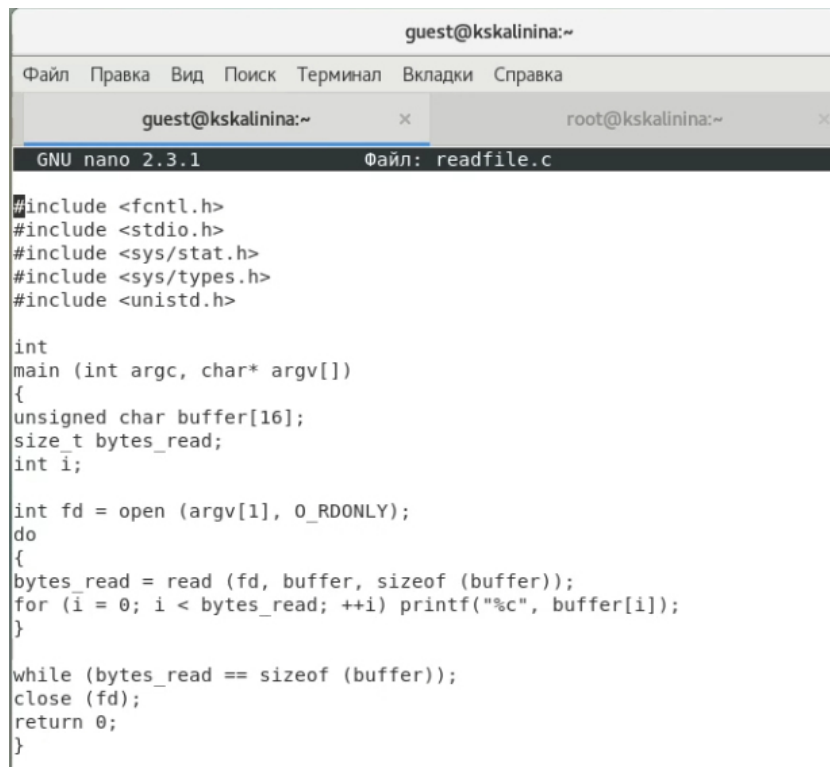
Figure 0.8: Установка атрибутов



```
[guest@kskalinina ~]$ ls -l simpleid2
-rwsrwsr-x. 1 root guest 8576 ноя 13 08:33 simpleid2
[guest@kskalinina ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@kskalinina ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

Figure 0.9: Повтор действий с SetGID-битом

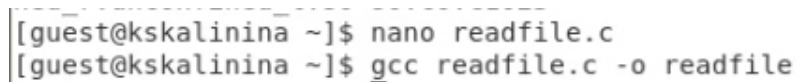
8. Создала программу readfile.c (fig. 0.10).



```
guest@kskalinina:~  
Файл Правка Вид Поиск Терминал Вкладки Справка  
guest@kskalinina:~ x root@kskalinina:~ x  
GNU nano 2.3.1 Файл: 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;  
}
```

Figure 0.10: Код программы readfile.c

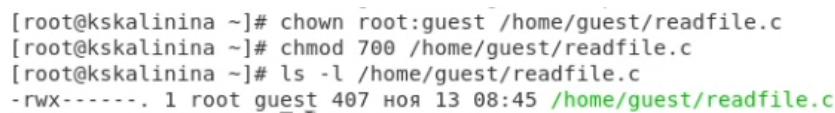
9. Откомпилировала её (fig. 0.11).



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

Figure 0.11: Компиляция программы readfile.c

10. Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (fig. 0.12).



```
[root@kskalinina ~]# chown root:guest /home/guest/readfile.c  
[root@kskalinina ~]# chmod 700 /home/guest/readfile.c  
[root@kskalinina ~]# ls -l /home/guest/readfile.c  
-rwx----- 1 root guest 407 ноя 13 08:45 /home/guest/readfile.c
```

Figure 0.12: Смена владельца файла readfile.c и прав на него

11. Убедилась, что guest не может прочитать файл readfile.c (fig. 0.13).

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

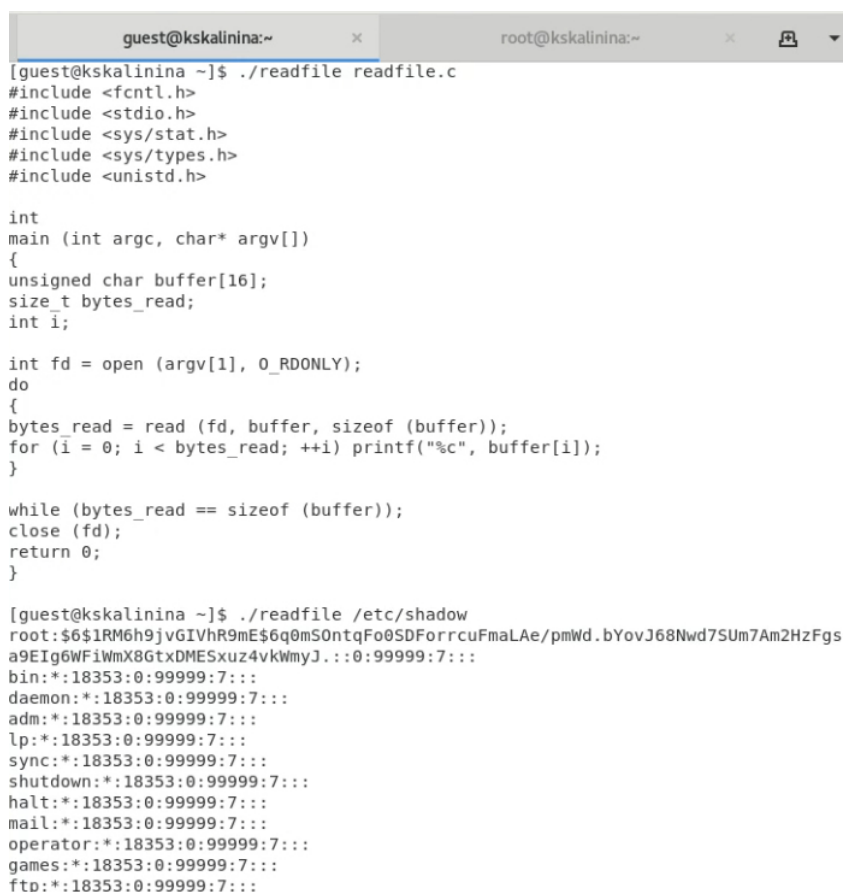
Figure 0.13: Отказ в чтении readfile.c

12. Смените у программы readfile владельца и установите SetU'D-бит (fig. 0.14).

```
[root@kskalinina ~]# chown root:guest /home/guest/readfile
[root@kskalinina ~]# chmod u+s /home/guest/readfile
[root@kskalinina ~]# ls -l /home/guest/readfile
-rwsrwxr-x. 1 root guest 8512 ноя 13 08:45 /home/guest/readfile
```

Figure 0.14: Смена владельца файла readfile и установка SetU'D-бит

13. Убедилась, что readfile может прочитать файлы readfile.c и “/etc/shadow” (fig. 0.15).



```
guest@kskalinina:~ x root@kskalinina:~ x
[guest@kskalinina ~]$ ./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@kskalinina ~]$ ./readfile /etc/shadow
root:$6$1RM6h9jvGIVhR9mE$6q0mS0ntqFo0SDForrcuFmaLAe/pmWd.bYovJ68Nwd7SUm7Am2HzFgs
a9EIg6WFiWmX8GtxDMESxuz4vkwmyJ.:0:99999:7:::
bin:!:18353:0:99999:7:::
daemon:!:18353:0:99999:7:::
adm:!:18353:0:99999:7:::
lp:!:18353:0:99999:7:::
sync:!:18353:0:99999:7:::
shutdown:!:18353:0:99999:7:::
halt:!:18353:0:99999:7:::
mail:!:18353:0:99999:7:::
operator:!:18353:0:99999:7:::
games:!:18353:0:99999:7:::
ftp:!:18353:0:99999:7:::
```

Figure 0.15: Чтение readfile.c и “/etc/shadow”

14. Убедилась, что атрибут Sticky установлен на директории “/tmp” (fig. 0.16).

```
[guest@kskalinina ~]$ ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 ноя 13 08:49 tmp
```

Figure 0.16: Атрибут Sticky на директории /tmp

15. От имени пользователя guest создала файл file01.txt в директории “/tmp” со словом test. Разрешила чтение и запись для категории пользователей «все остальные» (fig. 0.17).

```
[guest@kskalinina ~]$ echo "test" > /tmp/file01.txt
[guest@kskalinina ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 13 08:50 /tmp/file01.txt
[guest@kskalinina ~]$ chmod o+rw /tmp/file01.txt
[guest@kskalinina ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 13 08:50 /tmp/file01.txt
```

Figure 0.17: Создание file01.txt и смена атрибутов

16. От пользователя guest2 просмотрела файл, успешно дозаписала и переписала его. Но не смогла его удалить (fig. 0.18).

```
guest@kskalinina:~ x root@kskalinina:~ x guest2@kskalinina:/h... x
[guest@kskalinina ~]$ su guest2
Пароль:
[guest2@kskalinina guest]$ cat /tmp/file01.txt
test
[guest2@kskalinina guest]$ echo "test2" >> /tmp/file01.txt
[guest2@kskalinina guest]$ cat /tmp/file01.txt
test
test2
[guest2@kskalinina guest]$ echo "test3" > /tmp/file01.txt
[guest2@kskalinina guest]$ cat /tmp/file01.txt
test3
[guest2@kskalinina guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
```

Figure 0.18: Работа с file01.txt от guest2

17. От суперпользователя сняла атрибут t (Sticky-бит) с директории “/tmp” (fig. 0.19).

```
[root@kskalinina ~]# chmod -t /tmp
```

Figure 0.19: Снятие атрибута Sticky с директории /tmp

18. Убедилась в правильности снятия атрибута и повторила предыдущие шаги. В этот раз удаление также прошло успешно (fig. 0.20).

```
[guest2@kskalinina guest]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 ноя 13 08:54 tmp
[guest2@kskalinina guest]$ cat /tmp/file01.txt
test3
[guest2@kskalinina guest]$ echo "test2" >> /tmp/file01.txt
[guest2@kskalinina guest]$ cat /tmp/file01.txt
test3
test2
[guest2@kskalinina guest]$ echo "test" > /tmp/file01.txt
[guest2@kskalinina guest]$ cat /tmp/file01.txt
test
[guest2@kskalinina guest]$ rm /tmp/file01.txt
[guest2@kskalinina guest]$ ls /tmp
ssh-p4ccW0zsbZtc
systemd-private-3fdb085flef4488ea4acc18d5850a696-bolt.service-mmWSfg
systemd-private-3fdb085flef4488ea4acc18d5850a696-chronyd.service-eBERBM
systemd-private-3fdb085flef4488ea4acc18d5850a696-colord.service-coEaBA
systemd-private-3fdb085flef4488ea4acc18d5850a696-cups.service-g9hvGY
systemd-private-3fdb085flef4488ea4acc18d5850a696-fwupd.service-ENFI1L
systemd-private-3fdb085flef4488ea4acc18d5850a696-rtkit-daemon.service-anmI76
tracker-extract-files.1001
yum_save_tx.2021-11-13.08-15.DWWMJR.yumtx
```

Figure 0.20: Повторное выполнение команд от guest2

19. От суперпользователя вернула атрибут t (Sticky-бит) на директорию “/tmp” (fig. 0.21).

```
[root@kskalinina ~]# chmod +t /tmp
[root@kskalinina ~]# ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 ноя 13 08:56 tmp
```

Figure 0.21: Возвращение атрибута Sticky на директорию /tmp

Выводы

Таким образом я успешно приобрела изучила механизмы 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.

Список литературы

1. Использование SETUID, SETGID и Sticky bit. // ruvds.com 2021. URL: <https://ruvds.com/ru/helpcenter/suid-sgid-sticky-bit-linux/> (дата обращения 13.11.2021).
2. ИПрава в Linux (chown, chmod, SUID, GUID, sticky bit, ACL, umask). // habr.com 2019. URL: <https://habr.com/ru/post/469667/> (дата обращения 13.11.2021).
3. Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян. Информационная безопасность компьютерных сетей: лабораторные работы. // Факультет физико-математических и естественных наук. М.: РУДН, 2015. 64 с..