

Лабораторная работа № 5

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

Усов Александр Александрович НБибд-02-18

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Создание программы	8
4.2	Исследование Sticky-бита	13
5	Выводы	16
	Список литературы	17

Список иллюстраций

4.1	Создание программы	8
4.2	Программа	9
4.3	Команда <code>chmod 600 file1</code>	9
4.4	Усложненная программа	10
4.5	Скомпилировали и запустили	10
4.6	Выполнили команды	11
4.7	Перезапустили	11
4.8	<code>chmod g+s /home/guest/simpleid2</code>	11
4.9	<code>readfile.c</code>	12
4.10	<code>gcc readfile.c -o readfile</code>	12
4.11	может ли программа <code>readfile</code> прочитать файл <code>readfile.c</code>	13
4.12	Выяснили, установлен ли атрибут <code>Sticky</code> на директории	13
4.13	Действи с файлом	14
4.14	Снял атрибут <code>t</code>	15
4.15	Удалил файл	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.

2 Задание

1. Создать и проверить работу файла `simpleid.c`
2. Расширить работу файла `simpleid2.c` и проверить работу с разными атрибутами
3. Создать программу `readfile.c` и проверить атрибуты на нем.

3 Теоретическое введение

setuid и setgid являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами владельца или группы исполняемого файла. В Unix-подобных системах приложение запускается с правами пользователя, вызвавшего указанное приложение. Это обеспечивает дополнительную безопасность, так как процесс с правами пользователя не сможет получить доступ на запись к важным системным файлам, например /etc/passwd, который принадлежит суперпользователю root.

Более подробно см. в [1].

Sticky bit — дополнительный атрибут файлов или каталогов в операционных системах семейства UNIX. Впервые sticky bit появился в пятой редакции UNIX в 1974 году для использования в исполняемых файлах. Он применялся для уменьшения времени загрузки наиболее часто используемых программ. После закрытия программы код и данные оставались в памяти, а следующий запуск происходил быстрее

Более подробно см. в [2].

4 Выполнение лабораторной работы

4.1. Создание программы

1. Вошли в систему от имени пользователя guest.
2. Создали программу simpleid.c. Для этого запустили редактор vi с помощью команды vi simpleid.c:

(рис. 4.1):



```
[guest@aausov home]$ ls
aausov guest guest2
[guest@aausov home]$ cd guest
[guest@aausov ~]$ vi simpleid.c
[guest@aausov ~]$ ls
dirl  file2  Видео  Загрузки  Музыка  Рабочий стол
file  simpleid.c  Документы  Изображения  Общедоступные  Шаблоны
```

Рис. 4.1: Создание программы

Напечатали текст программы в открывшемся редакторе:(рис. 4.2).


```
#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. 4.2: Программа

3. Скомпилировали программу и убедились, что файл программы создав:
gcc simpleid.c -o simpleid

(рис. 4.3).

```
[guest@aauosv ~]$ gcc simpleid.c -o simpleid
[guest@aauosv ~]$ ls
dir1  file2  simpleid.c  Документы  Изображения  Общедоступные  Шаблоны
file  simpleid  Видео      Загрузки    Музыка       Рабочий стол
[guest@aauosv ~]$ ./simpleid
uid=1001, gid=1001
[guest@aauosv ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined u:unconfined r:unconfined t:s0-s0:c0.c1023
```

Рис. 4.3: Команда `chmod 600 file1`

4. Выполнили программу `simpleid` командой `./simpleid`

(рис. 4.3).

5. Выполнили системную программу `id` с помощью команды `id` (рис. 4.3). `uid` и `gid` совпадает в обеих программах

6. Усложнили программу, добавив вывод действительных идентификаторов.

Получившуюся программу назвали simpleid2.c. (рис. 4.4).

```
#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. 4.4: Усложненная программа

7. Скомпилировали и запустили simpleid2.c: gcc simpleid2.c -o simpleid2
./simpleid2

(рис. 4.5)

```
[guest@aausov ~]$ vi simpleid2.c
[guest@aausov ~]$ gcc simpleid2.c -o simpleid2
[guest@aausov ~]$ ls
dir1  simpleid  simpleid.c  Загрузки  Общедоступные
file  simpleid2  Видео      Изображения  Рабочий стол
file2 simpleid2.c  Документы  Музыка      Шаблоны
[guest@aausov ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Рис. 4.5: Скомпилировали и запустили

8. От имени суперпользователя выполнили команды: chown root:guest
/home/guest/simpleid2 chmod u+s /home/guest/simpleid2

(рис. 4.6)

```

[guest@aausov ~]$ su
Пароль:
[root@aausov guest]# chown root:guest /home/guest/simpleid2
[root@aausov guest]# chmod u+s /home/guest/simpleid2
[root@aausov guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8576 ноя  7 11:26 simpleid2

```

Рис. 4.6: Выполнили команды

9. Использовали su для повышения прав до суперпользователя
10. Выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2: `ls -l simpleid2` (рис. 4.6)
11. Запустили simpleid2 и id: `./simpleid2`, id Результат выполнения программ теперь немного отличается (рис. 4.7)

```

[guest@aausov ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aausov ~]$ su
Пароль:
[root@aausov guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@aausov guest]# id
uid=0(root) gid=0(root) rpyнны=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@aausov guest]# sudo ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@aausov guest]# sudo id
uid=0(root) gid=0(root) rpyнны=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

```

Рис. 4.7: Перезапустили

12. Проделали тоже самое относительно SetGID-бита. От имени суперпользователя выполним команду `chmod g+s /home/guest/simpleid2` Запустили simpleid2 и id (рис. 4.8)

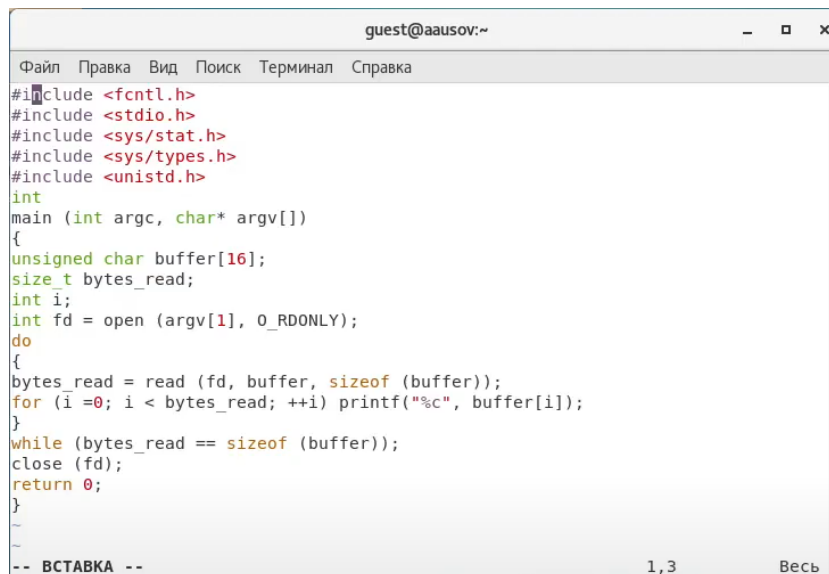
```

su: user ./simpleid2 does not exist
[root@aausov guest]# su guest
[guest@aausov ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 8576 ноя  7 11:26 simpleid2
[guest@aausov ~]$ su
Пароль:
[root@aausov guest]# chmod g+s /home/guest/simpleid2
[root@aausov guest]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 8576 ноя  7 11:26 simpleid2
[root@aausov guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@aausov guest]# id
uid=0(root) gid=0(root) rpyнны=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

```

Рис. 4.8: `chmod g+s /home/guest/simpleid2`


13. Создали программу readfile.c с помощью команды `vi readfile.c` (рис. 4.9)



```
guest@aasov:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
#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;  
}  
~  
~  
-- ВСТАВКА -- 1,3  Весь
```

Рис. 4.9: readfile.c

14. Откомпилировали её. (рис. 4.10) `gcc readfile.c -o readfile`



```
[guest@aasov ~]$ vi readfile.c  
[guest@aasov ~]$ gcc readfile.c -o readfile  
[guest@aasov ~]$ su  
Пароль:  
[root@aasov guest]# chown root readfile.c  
[root@aasov guest]# chmod 700 readfile.c  
[root@aasov guest]# su guest  
[guest@aasov ~]$ cat readfile.c  
cat: readfile.c: Отказано в доступе
```

Рис. 4.10: gcc readfile.c -o readfile

15. Сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. `chown root:guest /home/guest/readfile.c chmod 700 /home/guest/readfile.c` (рис. 4.10)
16. Проверили, что пользователь guest не может прочитать файл readfile.c. (рис. 4.10)
17. Сменили у программы readfile владельца и установили SetU'D-бит. (рис. 4.11)
18. Проверили, может ли программа readfile прочитать файл readfile.c (рис. 4.11)

23. От пользователя guest2 прочитать файл /tmp/file01.txt: cat /tmp/file01.txt (рис. 4.13)

```
[aausov@aausov ~]$ su
Пароль:
[root@aausov aausov]# su guest2
[guest2@aausov aausov]$ cat /tmp/file01.txt
test
[guest2@aausov aausov]$ cd ..
[guest2@aausov home]$ cat /tmp/file01.txt
test
[guest2@aausov home]$ echo "test2" > /tmp/file01.txt
[guest2@aausov home]$ cat /tmp/file01.txt
test2
[guest2@aausov home]$ echo "test3" > /tmp/file01.txt
[guest2@aausov home]$ cat /tmp/file01.txt
test3
[guest2@aausov home]$ echo "test2" >> /tmp/file01.txt
[guest2@aausov home]$ cat /tmp/file01.txt
test3
test2
[guest2@aausov home]$ echo "test3" > /tmp/file01.txt
[guest2@aausov home]$ cat /tmp/file01.txt
test3
[guest2@aausov home]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Нет такого файла или каталога
[guest2@aausov home]$
```

Рис. 4.13: Действия с файлом

24. От пользователя guest2 попробовали дозаписать в файл /tmp/file01.txt слово test3 командой: echo "test2" » /tmp/file01.txt (рис. 4.13)
25. Проверили содержимое файла командой: cat /tmp/file01.txt (рис. 4.13)
26. От пользователя guest2 попробовали записать в файл /tmp/file01.txt слово test4, стерев при этом всю имеющуюся в файле информацию командой. Для этого воспользовались командой echo "test3" > /tmp/file01.txt Проверили содержимое файла командой (рис. 4.13)
27. От пользователя guest2 попробовал удалить файл /tmp/file01.txt командой rm /tmp/file01.txt, однако получил отказ. (рис. 4.13)
28. От суперпользователя командой выполнили команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp (рис. 4.14)

```

[root@aausov ~]# chmod -t /tmp
[root@aausov ~]# exit
logout
[guest2@aausov home]$ ls -l / | grep tmp
drwxrwxrwx. 19 root root 4096 ноя 7 11:52 tmp
[guest2@aausov home]$ cat /tmp/file01.txt
test3
[guest2@aausov home]$ echo "test2" >> /tmp/file01.txt
[guest2@aausov home]$ cat /tmp/file01.txt
test3
test2
[guest2@aausov home]$ echo "test3" > /tmp/file01.txt
[guest2@aausov home]$ cat /tmp/file01.txt
test3

```

Рис. 4.14: Снял атрибут t

29. От пользователя guest2 проверили, что атрибута t у директории /tmp нет:
ls -l / | grep tmp (рис. 4.14)
30. Повторили предыдущие шаги. (рис. 4.14) Получилось удалить файл (рис. 4.15)

```

[guest2@aausov home]$ rm /tmp/file01.txt
[guest2@aausov home]$ su
Пароль:
[root@aausov home]# chmod +t /tmp
[root@aausov home]# exit
exit
[guest2@aausov home]$

```

Рис. 4.15: Удалил файл

31. Повысили свои права до суперпользователя и вернули атрибут t на директорию /tmp : su chmod +t /tmp exit (рис. 4.15)

5 Выводы

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

Список литературы

1. suid [Электронный ресурс]. Сайт, 2021. URL: <https://ru.wikipedia.org/wiki/Suid>.
2. Sticky bit [Электронный ресурс]. Сайт, 2021. URL: https://ru.wikipedia.org/wiki/Sticky_bit.