

Отчет по лабораторной работе №5

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

Дмитрий Сергеевич Шестаков

Содержание

1	Цель работы	5
2	Выполнение работы	6
3	Выводы	12
	Список литературы	13

Список иллюстраций

2.1	Программа simpleid.c	6
2.2	Запуск программы simpleid.c	6
2.3	Программа simpleid2.c	7
2.4	Компиляция программы simpleid2.c	7
2.5	Изменение атрибутов simpleid2	7
2.6	SetGID-бита	8
2.7	Программа readfile.c	8
2.8	Изменение пользователя	9
2.9	Установка SetUD-бит	9
2.10	Sticky-бит	10
2.11	Файл file01.txt	10
2.12	Чтение, запись, удаление file01.txt	10
2.13	Снятие Sticky-бит	11
2.14	Операции после снятия Sticky-бита	11

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 Выполнение работы

1. Зашли под именем пользователя guest. Создали программу simpleid.c. (рис. 2.1)

```
#include <unistd.h>
#include <sys/types.h>
#include <stdio.h>

int main() {
    uid_t uid = getuid();
    gid_t gid = getgid();
    printf("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

Рис. 2.1: Программа simpleid.c

2. Скомпилировали и запустили ее. Сравнили ее результат с выводом команды id. (рис. 2.2)

```
[guest@Rocky ~]$ vim simpleid.c
[guest@Rocky ~]$ vim simpleid.c
[guest@Rocky ~]$ ls
dir1 simpleid.c
[guest@Rocky ~]$ gcc simpleid.c -o simpleid
simpleid.c:8: функция «main»:
simpleid.c:8:22: ошибка: выборка элемента «uid» из объекта, не являющегося структурой или объединением
   8 |     printf("uid=%d, gid=%d\n", uid, gid);
     |                               ^
[guest@Rocky ~]$ vim simpleid.c
[guest@Rocky ~]$ gcc simpleid.c -o simpleid
[guest@Rocky ~]$ ls
dir1 simpleid simpleid.c
[guest@Rocky ~]$ ./simpleid
uid=1001, gid=1001
[guest@Rocky ~]$ id
uid=1001(guest) gid=1001(guest) rpnw=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@Rocky ~]$
```

Рис. 2.2: Запуск программы simpleid.c

3. Усложнили программу simpleid.c и получили программу simpleid2.c. (рис. 2.3)

```

#include <unistd.h>
#include <sys/types.h>
#include <stdio.h>

int main() {
    uid_t e_uid = geteuid();
    uid_t real_uid = getuid();

    gid_t e_gid = getegid();
    gid_t real_gid = getgid();

    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. 2.3: Программа simpleid2.c

4. Скомпилировали и запустили программу simpleid2.c (рис. 2.4)

```

[guest@Rocky ~]$ gcc simpleid2.c -o simpleid2
gcc: ошибка: ввод со стандартного ввода возможен только с кл
[guest@Rocky ~]$ gcc simpleid2.c -o simpleid2
[guest@Rocky ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001

```

Рис. 2.4: Компиляция программы simpleid2.c

5. От имени суперпользователя выполнили команды:

`chown root:guest /home/guest/simpleid2`

`chmod u+s /home/guest/simpleid2`

Выполнили проверку правильности установки новых атрибутов и запустили программу simpleid2.c и id (рис. 2.5)

```

[guest@Rocky ~]$ su - root
Пароль:
[root@Rocky ~]# chown root:guest /home/guest/simpleid2
[root@Rocky ~]# chmod u+s /home/guest/simpleid2
[root@Rocky ~]# su - guest
[guest@Rocky ~]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 26064 сен 23 11:03 simpleid2
[guest@Rocky ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@Rocky ~]$ id
uid=1001(guest) gid=1001(guest) rpyны=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@Rocky ~]$

```

Рис. 2.5: Изменение атрибутов simpleid2

6. Проделали то же самое относительно SetGID-бита (рис. 2.6)

```
guest@Rocky ~]$ su - root
root@Rocky: ~/# chmod g+s /home/guest/simpleid2
root@Rocky: ~/# su - guest
guest@Rocky ~]$ ls -l
total 60
drwx----- 2 guest guest 19 сен 23 08:33 dir1
-rwxr-xr-x 1 guest guest 25960 сен 23 10:59 simpleid
-rwsr-sr-x 1 root guest 26064 сен 23 11:03 simpleid2
-rw-r--r-- 1 guest guest 314 сен 23 11:02 simpleid2.c
guest@Rocky ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
guest@Rocky ~]$ id
uid=1001(guest) gid=1001(guest) rпгпы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
guest@Rocky ~]$
```

Рис. 2.6: SetGID-бита

7. Создали программу readfile.c (рис. 2.7)

```
#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. 2.7: Программа readfile.c

8. Изменили владельца файла readfile.c и настроили так, чтобы только супер-пользователь мог его прочитать. И проверили, что пользователь guest не может его прочитать. (рис. 2.8)


```

[root@Rocky ~]# chown root /home/guest/readfile.c
[root@Rocky ~]# chmod 700 /home/guest/readfile.c
[root@Rocky ~]# su - guest
[guest@Rocky ~]$ ls -l
итого 92
drwx-----. 2 guest guest   19 сен 23 08:33 dir1
-rwxr-xr-x. 1 guest guest 26008 сен 23 11:17 readfile
-rwx-----. 1 root guest   424 сен 23 11:16 readfile.c
-rwxr-xr-x. 1 guest guest 25960 сен 23 10:59 simpleid
-rwsr-sr-x. 1 root guest 26064 сен 23 11:03 simpleid2
-rw-r--r--. 1 guest guest   314 сен 23 11:02 simpleid2.c
[guest@Rocky ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@Rocky ~]$

```

Рис. 2.8: Изменение пользователя

9. Сменили у программы readfile пользователя и установили SetUD-бит. И проверили, что программа может прочитать файл readfile.c и /etc/shadow. (рис. 2.9)

```

[guest@Rocky ~]$ ./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@Rocky ~]$ ./readfile /etc/shadow
root:$6$87XXDTKI/294sx9R$EXW16mEu01H6ebdnuUn0KBWLyA8EFTXSz0gsgspMLhaX2NRqpGTNBrLT1PAHCLYSTD23UKD3rGzEooyrz4tE...:0:99999:7:::
bin:!:19469:0:99999:7:::
daemon:!:19469:0:99999:7:::
adm:!:19469:0:99999:7:::
lp:!:19469:0:99999:7:::
sync:!:19469:0:99999:7:::

```

Рис. 2.9: Установка SetUD-бит

10. Выяснили установлен ли Sticky-бит на директории /tmp командой (рис. 2.10):

```
ls -l / | grep /tmp
```

```
[guest@Rocky ~]$ su - root
Пароль:
[root@Rocky ~]# ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 сен 23 11:29 tmp
```

Рис. 2.10: Sticky-бит

11. Создали file01.txt, записали туда слово “test”, проверили атрибуты файла и разрешили чтение и запись для категории “все остальные”. (рис. 2.11)

```
[guest@Rocky ~]$ su - root
Пароль:
[root@Rocky ~]# ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 сен 23 11:29 tmp
[root@Rocky ~]# su - guest
[guest@Rocky ~]$ echo "test" > /tmp/file01.txt
[guest@Rocky ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 сен 23 11:30 /tmp/file01.txt
[guest@Rocky ~]$ chmod o+rw /tmp/file01.txt
[guest@Rocky ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 сен 23 11:30 /tmp/file01.txt
```

Рис. 2.11: Файл file01.txt

12. Попытались прочитать файл, дописать, переписать и удалить файл от имени пользователя guest2. Получилось только прочитать file01.txt (рис. 2.12)

```
[guest@Rocky ~]$ su - guest2
Пароль:
[guest2@Rocky ~]$ echo "test2" > /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@Rocky ~]$ cat /tmp/file01.txt
test
[guest2@Rocky ~]$ echo "test3" >> /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@Rocky ~]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
```

Рис. 2.12: Чтение, запись, удаление file01.txt

13. Перешли в режим суперпользователя и сняли Sticky-бит. (рис. 2.13)

```
[guest2@Rocky ~]$ su -
Пароль:
[root@Rocky ~]# chmod -t /tmp
[root@Rocky ~]# exit
Выход
[guest2@Rocky ~]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 сен 23 11:37 tmp
```

Рис. 2.13: Снятие Sticky-бит

14. Проделали все те же шаги из пункта 12. В этот раз также удалось удалить файл. (рис. 2.14)

```
guest2@Rocky ~]$ cat /tmp/file01.txt
test
guest2@Rocky ~]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
guest2@Rocky ~]$ cat /tmp/file01.txt
test
guest2@Rocky ~]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
guest2@Rocky ~]$ rm /tmp/file01.txt
m: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
guest2@Rocky ~]$
```

Рис. 2.14: Операции после снятия Sticky-бита

3 Выводы

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

Список литературы

1. Кулябов Д.С., Королькова А. В., Геворкян М. Н. Лабораторная работа No 5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов [Электронный ресурс]. URL: https://esystem.rudn.ru/pluginfile.php/2090208/mod_resource/content/2/005-lab_discret_sticky.pdf.