

Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Шестаков Д. С.

30 сентября 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Шестаков Дмитрий Сергеевич
- студент группы НКНбд-01-20
- Российский университет дружбы народов
- dmshestakov@icloud.com

Вводная часть

- Дискрецинное разграничение прав
- ОС Linux
- Bash

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

- ОС Linux
- Bash

Зашли под именем пользователя guest. Создали программу simpleid.c.

```
#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. 1: Программа simpleid.c

Скомпилировали и запустили ее. Сравнили ее результат с выводом команды `id`

```
[guest@Rocky ~]$ vim simpleid.c
[guest@Rocky ~]$ vim simpleid.c
[guest@Rocky ~]$ ls
lrwxrwxrwx. simpleid.c
[guest@Rocky ~]$ gcc simpleid.c -o simpleid
simpleid.c: в функции «main»:
simpleid.c:8:28: 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
lrwxrwxrwx. simpleid simpleid.c
[guest@Rocky ~]$ ./simpleid
uid=1001, gid=1001
[guest@Rocky ~]$ id
uid=1001(guest) gid=1001(guest) rpyнм=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@Rocky ~]$
```

Рис. 2: Запуск программы `simpleid.c`

Усложнили программу simpleid.c и получили программу simpleid2.c

```
#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. 3: Программа simpleid2.c

Скомпилировали и запустили программу simpleid2.c

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

Рис. 4: Компиляция программы simpleid2.c

От имени суперпользователя выполнили команды:

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

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

Выполнили проверка правильности установки новых атрибутов и запустили программу simpleid2.c и id

```
[guest@Rocky ~]$ su - root
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) rpnmu=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@Rocky ~]$
```

Рис. 5: Изменение атрибутов simpleid2

Проделили то же самое относительно SetGID-бита

```
guest@Rocky ~]$ su - root
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
uid=0, e_gid=1001
real_uid=1001, real_gid=1001
guest@Rocky ~]$ id
uid=1001(guest) gid=1001(guest) rpynmw=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
guest@Rocky ~]$
```

Рис. 6: SetGID-бита

Создали программу 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;
}
```

Рис. 7: Программа readfile.c

Изменили владельца файла readfile.c и настроили так, чтобы только суперпользователь мог его прочитать. И проверили, что пользователь guest не может его прочитать.

```
[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. 8: Изменение пользователя

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

```
[guest@Rocky ~]$ ./readfile readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

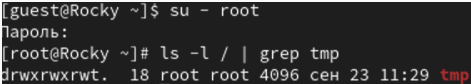
int main(int argc, char* argv[]) {
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    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$87xxDTK1/20ax0R$EXM16mU0IH6ebdnU0K0WLy0EFTX5z0gsgspMLhaX2NRqpGTN0RLT1PANCLYS7D23KD3rG2E0dyrz4tE.: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. 9: Установка SetUD-бит

Выяснили установлен ли Sticky-бит на директории /tmp командой (рис. (fig:010?)):

```
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. 10: Sticky-бит

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

```
[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
```

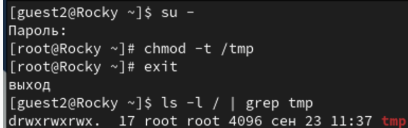
Рис. 11: Файл file01.txt

Попытались прочитать файл, дописать, переписать и удалить файл от имени пользователя guest2. Получилось только прочитать file01.txt

```
[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. 12: Чтение, запись, удаление file01.txt

Перешли в режим суперпользователя и сняли Sticky-бит.



```
[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. 13: Снятие Sticky-бит

Проделили все те же шаги из пункта 12. В этот раз также удалось удалить файл.

```
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
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
guest2@Rocky ~]$
```

Рис. 14: Операции после снятия Sticky-бита

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.