

Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Коняева Марина Александровна

НФИбд-01-21

Студ. билет: 1032217044

2024

RUDN

Дискреционное разграничение доступа — управление доступом субъектов к объектам на основе списков управления доступом или матрицы доступа. Также используются названия дискреционное управление доступом, контролируемое управление доступом и разграничительное управление доступом. [2]

SetUID

setuid и setgid (сокращения от англ. set user ID upon execution — «установка ID пользователя во время выполнения» и англ. set group ID upon execution — «установка ID группы во время выполнения») являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами владельца или группы исполняемого файла. [3]

Sticky

Sticky bit используется в основном для каталогов, чтобы защитить в них файлы. Из такого каталога пользователь может удалить только те файлы, владельцем которых он является. Примером может служить каталог /tmp, в который запись открыта для всех пользователей, но нежелательно удаление чужих файлов. [4]

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.

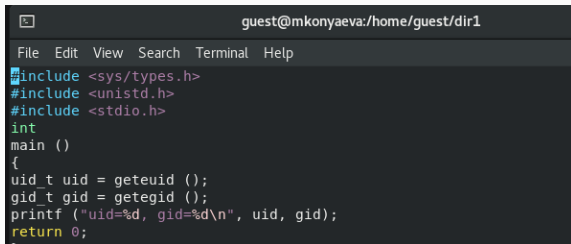
```
[guest@mkonyaeva ~]$ su
Password:
[root@mkonyaeva guest]# yum install gcc
Rocky Linux 8 - AppStream                2.1 kB/s | 4.8 kB    00:02
Rocky Linux 8 - AppStream                358 kB/s | 13 MB    00:36
Rocky Linux 8 - BaseOS                  289 kB/s | 7.2 MB    00:25
Rocky Linux 8 - Extras                   3.1 kB/s | 14 kB    00:04
Package gcc-8.5.0-22.el8_10.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@mkonyaeva guest]# setenforce 0
[root@mkonyaeva guest]# getenforce
Permissive
[root@mkonyaeva guest]#
```

Рис. 1: (рис. 1. Установка gss)

1. Зашли в систему от имени пользователя guest.
2. Создали файл simpleid.c, записали в него программу, скопировали и запустили его. Программа дала те же результаты, что и консольная команда id. (@fig:001, @fig:002)

```
[guest@mkonyaeva dir1]$ touch simpleid1.c
[guest@mkonyaeva dir1]$ vim simpleid1.c
[guest@mkonyaeva dir1]$ gcc simpleid1.c -o simpleid1
[guest@mkonyaeva dir1]$ ./simpleid1
uid=1001, gid=1001
[guest@mkonyaeva dir1]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@mkonyaeva dir1]$
```

Рис. 2: Работа в консоли с файлом simpleid.c

A screenshot of a code editor window. The title bar shows the path 'guest@mkonyaeva:/home/guest/dir1'. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The code is written in C and includes headers for `<sys/types.h>`, `<unistd.h>`, and `<stdio.h>`. It defines an `int` type and a `main` function. Inside `main`, it declares `uid_t uid` and `gid_t gid`, gets their values using `geteuid` and `getegid`, prints them with `printf`, and returns 0.

```
guest@mkonyaeva:/home/guest/dir1
File Edit View Search Terminal Help
#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. 3: Содержимое файла simpleid.c

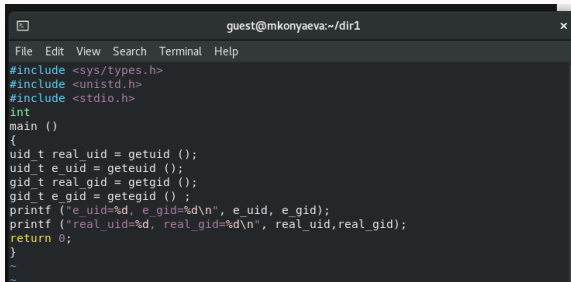
3. Создали файл simpleid2.c, записали в него программу, скопировали и запустили его. (@fig:003, @fig:004)

```
[guest@mkonyaeva dir1]$ ls
file1 file2 simpleid1 simpleid1.c simpleid2.c test1
[guest@mkonyaeva dir1]$ vim simple2.c
[guest@mkonyaeva dir1]$ vim simpleid2.c

[5]+ Stopped vim simpleid2.c
[guest@mkonyaeva dir1]$ gcc simpleid2.c -o simpleid2
[guest@mkonyaeva dir1]$ ./simpleid2
e_uid=1001, e_gid=1001
real uid=1001, real gid=1001
[guest@mkonyaeva dir1]$
```

Рис. 4: Работа в консоли с файлом simpleid2.c

Содержимое файла simpleid2.c



```
guest@mkonyaeva:~/dir1
File Edit View Search Terminal Help
#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. 5: Содержимое файла simpleid2.c

4. Изменили права файла simpleid2 от имени суперпользователя.
(@fig:005)

```
[root@mkonyaeva dir1]# chown root:guest /home/guest/dir1/simpleid2  
[root@mkonyaeva dir1]# chmod u+s /home/guest/dir1/simpleid2
```

Рис. 6: Изменение прав файла simpleid2

5. Выполнили проверку установки правил. Запустили simpleid2 и id. Получили одинаковы результаты с id=0. (@fig:006)

```
[root@mkonyaeva dir1]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 18312 Oct  5 02:06 simpleid2
[root@mkonyaeva dir1]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@mkonyaeva dir1]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
[root@mkonyaeva dir1]#
```

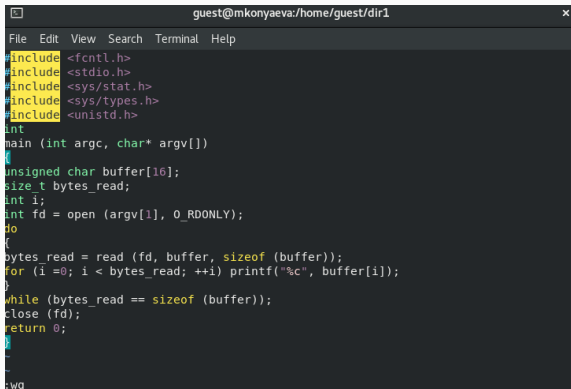
Рис. 7: Проверка прав файла simpleid2, его запуск и команда id

6. Повторили п.5 для SetGID-бита. (@fig:007)

Выполнения файла с SetGID-битом

Рис. 8: Выполнения файла с SetGID-битом

7. Создали программу readfile.c и откомпилировали ее. (@fig:008, @fig:009)

A screenshot of a terminal window with a dark background. The title bar shows the user 'guest' at 'mkonyaeva:/home/guest/dir1'. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The code is written in a light-colored font with syntax highlighting: preprocessor directives are blue, keywords are green, and string literals are red. The code defines a main function that opens a file specified by argv[1] in read-only mode, reads its contents into a 16-byte buffer, and prints each byte on a new line. The code ends with a return statement and a closing brace for the main function.

```
guest@mkonyaeva:/home/guest/dir1
File Edit View Search Terminal Help
#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;
}
:wq
```

Рис. 9: Содержимое файла readfile.c

```
[root@mkonyaeva dir1]# touch readfile.c  
[root@mkonyaeva dir1]# vim readfile.c  
[root@mkonyaeva dir1]# gcc readfile.c -o readfile
```

Рис. 10: Создание и компелирование readfile.c

8. Изменили права так, чтобы только суперпользователь (root) мог прочитать readfile.c, а guest не мог. (@fig:010)

```
[root@mkonyaeva dir1]# chown root:guest /home/guest/dir1/readfile.c  
[root@mkonyaeva dir1]# chmod 700 /home/guest/dir1/readfile.c  
[root@mkonyaeva dir1]#
```

Рис. 11: Изменение прав файла readfile.c

9. Проверили, что guest не может прочитать файл. (@fig:011)

```
[guest@mkonyaeva dir1]$ cat readfile.c  
cat: readfile.c: Permission denied
```

Рис. 12: Чтение readfile.c пользователем guest

10. Сменили у программы readfile владельца и установили SetU'D-бит. (@fig:012)

```
[guest@mkonyaeva dir1]$ su
Password:
[root@mkonyaeva dir1]# chown root:guest /home/guest/dir1/readfile
[root@mkonyaeva dir1]# chmod u+s /home/guest/dir1/readfile
[root@mkonyaeva dir1]#
```

Рис. 13: Смена прав у readfile

11. Считали программой readfile readfile.c и /etc/shadow. (@fig:013, @fig:014)

```
[root@mkonyaeva dir1]# ./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;
}
[root@mkonyaeva dir1]# ./readfile /etc/shadow
root:$6$Nde7ru3sWT7Zt/Xt$UmZyw3uk.C0Dct9Ufq0Fsf7fio36TKX00SAgb2L4u1X.An6B6JUSP.
```

Рис. 14: Чтение readfile.c через readfile

Чтение /etc/shadow через readfile

```
[root@mkonyaeva dir1]# ./readfile /etc/shadow
root:$6$Nde7ru3swT7Zt/Xt$UmZyw3uk.C0Dct9Ufq0Fsfn7fio36TKX00SAgb2L4u1X.An6B6JUSP.
dLY6uMsVznALN0ab/MhCqwyV21NHUG1::0:99999:7:::
bin:!:19767:0:99999:7:::
daemon:!:19767:0:99999:7:::
adm:!:19767:0:99999:7:::
lp:!:19767:0:99999:7:::
sync:!:19767:0:99999:7:::
shutdown:!:19767:0:99999:7:::
halt:!:19767:0:99999:7:::
mail:!:19767:0:99999:7:::
operator:!:19767:0:99999:7:::
games:!:19767:0:99999:7:::
ftp:!:19767:0:99999:7:::
nobody:!:19767:0:99999:7:::
dbus:!!:19985:!!!!:
systemd-coredump:!!:19985:!!!!:
systemd-resolve:!!:19985:!!!!:
tss:!!:19985:!!!!:
polkitd:!!:19985:!!!!:
geoclue:!!:19985:!!!!:
unbound:!!:19985:!!!!:
rtkit:!!:19985:!!!!:
pipewire:!!:19985:!!!!:
dnsmasq:!!:19985:!!!!:
clevis:!!:19985:!!!!:
usbmuxd:!!:19985:!!!!:
gluster:!!:19985:!!!!:
rpc:!!:19985:0:99999:7:::
chrony:!!:19985:!!!!:
saslauth:!!:19985:!!!!:
libstoragemgmt:!!:19985:!!!!:
sssd:!!:19985:!!!!:
avahi:!!:19985:!!!!:
```

Рис. 15: Чтение /etc/shadow через readfile

Исследование Sticky-бита. Создание и изменение прав файла /tmp/file01.txt

1. Проверили установлены ли на директории tmp атрибут Sticky. От имени пользователя guest создали file01.txt в директории /tmp со словом test. Просмотрели атрибуты у файла и разрешили чтение и запись для категории пользователей «все остальные». (@fig:015)

```
[guest@mkonyaeva dir1]$ ls -l / | grep tmp
drwxrwxrwt. 11 root root 4096 Oct  5 02:20 tmp
[guest@mkonyaeva dir1]$ echo "test" > /tmp/file01.txt
[guest@mkonyaeva dir1]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Oct  5 02:25 /tmp/file01.txt
[guest@mkonyaeva dir1]$ chmod o+rw /tmp/file01.txt
[guest@mkonyaeva dir1]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Oct  5 02:25 /tmp/file01.txt
[guest@mkonyaeva dir1]$
```

Рис. 16: Создание и изменение прав файла /tmp/file01.txt

2. От имени пользователя guest2 попробовали прочитать, дозаписать, переписать и удалить файл file01.txt. (@fig:016)


```
[guest@mkonyaeva dir1]$ su guest2
Password:
[guest2@mkonyaeva dir1]$ cat /tmp/file01.txt
test
[guest2@mkonyaeva dir1]$ echo "test2" > /tmp/file01.txt
[guest2@mkonyaeva dir1]$ cat /tmp/file01.txt
test2
[guest2@mkonyaeva dir1]$ echo "test3" > /tmp/file01.txt
[guest2@mkonyaeva dir1]$ cat /tmp/file01.txt
test3
[guest2@mkonyaeva dir1]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': No such file or directory
[guest2@mkonyaeva dir1]$
```

Рис. 17: Взаимодействие с file01.txt пользователем guest2 с Sticky-bit

3. Суперпользователем сняли Sticky-bit с каталога tmp. Повторили действия с файлом из п.2. (@fig:017)

![Взаимодействие с file01.txt пользователем

4. Вернули каталогу tmp Sticky-bit суперпользователем. (@fig:018)



```
[guest2@mkonyaeva dir1]$ su -  
Password:  
[root@mkonyaeva ~]# chmod +t /tmp  
[root@mkonyaeva ~]# exit  
logout  
[guest2@mkonyaeva dir1]$
```

Рис. 18: Возврат Sticky-bit каталогу tmp

В ходе выполнения лабораторной работы были опробованы действия на практике SetUID- и Sticky-битов и рассмотрен механизм смены идентификатора процессов пользователей.

[1] Методические материалы курса.

[2] Wikipedia: Избирательное управление доступом. (URL: <https://ru.wikipedia.org/wiki/%D0%98%D0%B7%D0%B1%D0%B8%D1%80%D0%B0>)

[3] Wikipedia: suid (URL: <https://ru.wikipedia.org/wiki/Suid>)

[4] Wikipedia: Sticky bit (URL: https://ru.wikipedia.org/wiki/Sticky_bit)4.