

# Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния  
дополнительных атрибутов.

Ли Тимофей Александрович

# Содержание

Цель работы	5
Выполнение лабораторной работы	6
Выводы	14

## Список таблиц

## Список иллюстраций

0.1	установка gcc . . . . .	6
0.2	создание simpleid.c . . . . .	7
0.3	simpleid.c . . . . .	7
0.4	simpleid2.c . . . . .	8
0.5	работа с SetUID и SetGID битами . . . . .	9
0.6	readfile.c . . . . .	10
0.7	действия с readfile . . . . .	11
0.8	чтение etc/shadow . . . . .	12
0.9	действия с Sticky-битом . . . . .	13

## 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.

# Выполнение лабораторной работы

Для начала я, зайдя в root, установил компилятор gcc со всеми сопутствующими пакетами. (рис. @fig:001):

```
[guest@localhost ~]$ yum install gcc
Ошибка: Эту команду нужно запускать с привилегиями суперпользователя (на большинстве систем - под именем пользователя root).
[guest@localhost ~]$ su root
Пароль:
[root@localhost guest]# yum install gcc
CentOS Linux 8 - AppStream                0.0 B/s | 0 B    00:00
Errors during downloading metadata for repository 'appstream':
- Curl error (6): Couldn't resolve host name for http://mirrorlist.centos.org/?release=8&arch=x86_64&repo=AppStream&infra=stock [Could not resolve host: mirrorlist.centos.org]
Ошибка: Не удалось загрузить метаданные для репозитория «appstream»: Cannot prepare internal mirrorlist: Curl error (6): Couldn't resolve host name for http://mirrorlist.centos.org/?release=8&arch=x86_64&repo=AppStream&infra=stock [Could not resolve host: mirrorlist.centos.org]
[root@localhost guest]# yum install gcc
CentOS Linux 8 - AppStream                8.2 kB/s | 4.3 kB    00:00
CentOS Linux 8 - AppStream                5.9 MB/s | 9.6 MB    00:01
CentOS Linux 8 - BaseOS                    6.8 kB/s | 3.9 kB    00:00
CentOS Linux 8 - BaseOS                    5.3 MB/s | 8.5 MB    00:01
CentOS Linux 8 - Extras                    2.9 kB/s | 1.5 kB    00:00
Зависимости разрешены.
=====
Пакет                Архитектура          Версия                Репозиторий           Размер
=====
Установка:
gcc                  x86_64                8.4.1-1.el8           appstream              23 М
Установка зависимостей:
cpp                  x86_64                8.4.1-1.el8           appstream              10 М
glibc-devel          x86_64                2.28-151.el8          baseos                 1.0 М
glibc-headers         x86_64                2.28-151.el8          baseos                 478 к
isl                   x86_64                0.16.1-6.el8          appstream              841 к
kernel-headers        x86_64                4.18.0-305.25.1.el8_4 baseos                 7.2 М
libxcrypt-devel       x86_64                4.1.1-4.el8           baseos                 25 к
=====
Результат транзакции
=====
Установка 7 Пакетов

Объем загрузки: 43 М
Объем изменений: 98 М
Продолжить? [д/н]: д
Загрузка пакетов:
CentOS Linux 8 - BaseOS                    194% [=====]
(1/7): isl-0.16.1-6.el8.x86_64.rpm          2.9 MB/s | 841 kB    00:00
(2/7): glibc-devel-2.28-151.el8.x86_64.rpm  1.8 MB/s | 1.0 MB    00:00
(3/7): glibc-headers-2.28-151.el8.x86_64.rpm 6.0 MB/s | 478 kB    00:00
(4/7): kernel-headers-4.18.0-305.25.1.el8_4.x86_64.rpm 9.1 MB/s | 7.2 MB    00:00
(5/7): libxcrypt-devel-4.1.1-4.el8.x86_64.rpm 238 kB/s | 25 kB     00:00
```

Рис. 0.1: установка gcc

Затем, установил setenforce 0, проверил запуск gcc и g++. Создал файл simpleid.c, скомпилировал его и запустил. Сравнил вывод с выводом команды id: (рис. @fig:002)

```

[root@localhost guest]# su guest
[guest@10 ~]$ pwd
/home/guest
[guest@10 ~]$ setenforce 0
setenforce: setenforce() failed
[guest@10 ~]$ su root
Пароль:
[root@10 guest]# setenforce 0
[root@10 guest]# getenforce
Permissive
[root@10 guest]# su guest
[guest@10 ~]$ whereid gcc
bash: whereid: команда не найдена...
[guest@10 ~]$ whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/libexec/gcc /usr/share/man/man1/gcc.1.gz /usr/share/info/gcc.info.gz
[guest@10 ~]$ whereis g++
g++:
[guest@10 ~]$ gcc -c file.c
gcc: ошибка: file.c: Нет такого файла или каталога
gcc: фатальная ошибка: не заданы входные файлы
компиляция прервана.
[guest@10 ~]$ touch simpleid.c
[guest@10 ~]$ gcc simpleid.c -o simpleid
[guest@10 ~]$ ./simpleid
uid=1001, gid=1001
[guest@10 ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@10 ~]$

```

Рис. 0.2: создание simpleid.c

Код файла simpleid.c: (рис. @fig:003)

```

#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. 0.3: simpleid.c

Как видим, вывод программы и команды `id` совпадают.

Добавил в код вывод действительных идентификаторов и сохранил файл как `simpleid2.c`: (рис. @fig:004)

```

#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. 0.4: simpleid2.c

Скомпилировал код и от имени root изменил владельца полученной программы на root и группу guest, а также добавил владельцу программы право на выполнение от имени суперпользователя. Затем, убедился в правильности установки атрибутов с помощью `ls -l`, запустил программу и вызвал команду `id`. Далее, с помощью root установил на программу для группы право выполнения от имени суперпользователя (SetGID бит) и повторил запуск программы (конец действий на скриншоте 6): (рис. @fig:005)



```

[guest@10 ~]$ gcc simpleid2.c -o simpleid2
simpleid2.c: В функции «main»:
simpleid2.c:12:16: ошибка: «getegid» не описан (первое использование в этой функции); имелось в виду «getgid»?
    gid_t e_gid = getegid ();
                   ^~~~~~
                   getegid
simpleid2.c:12:16: замечание: сообщение о каждом неопisanном идентификаторе выдается один раз в каждой функции, где он встречается
simpleid2.c:12:24: ошибка: expected «,» or «;» before «)» token
    gid_t e_gid = getegid );
                   ^
[guest@10 ~]$ gcc simpleid2.c -o simpleid2
[guest@10 ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real uid=1001, real_gid=1001
[guest@10 ~]$ su root
Пароль:
[root@10 guest]# chown root:guest /home/guest/simpleid2
[root@10 guest]# chmod u+s /home/guest/simpleid2
[root@10 guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 17648 ноя 11 16:55 simpleid2
[root@10 guest]# ./simpleid2
e_uid=0, e_gid=0
real uid=0, real_gid=0
[root@10 guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@10 guest]# su guest
[guest@10 ~]$ ./simpleid2
e_uid=0, e_gid=1001
real uid=1001, real_gid=1001
[guest@10 ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@10 ~]$ chmod g+s /home/guest/simpleid2
chmod: изменение прав доступа для '/home/guest/simpleid2': Операция не позволена
[guest@10 ~]$ su root
Пароль:
[root@10 guest]# chmod g+s /home/guest/simpleid2
[root@10 guest]# su guest
[guest@10 ~]$ ./simpleid2
e_uid=0, e_gid=1001
real uid=1001, real_gid=1001

```

Рис. 0.5: работа с SetUID и SetGID битами

Затем, написал программу readfile и скомпилировал ее: (рис. @fig:006)

```

#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>
#include <fcntl.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. 0.6: readfile.c

Далее Сменил владельца программы на root и сменил права так, что только суперпользователь может читать этот файл. Убедился, что guest не может прочитав файл. Сменил владельца у скомпилированной программы и установил SetUID бит. Попробовал от имени guest прочитав файл readfile.c с помощью программы: (рис. @fig:007)

```

[guest@10 ~]$ su root
Пароль:
[root@10 guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@10 guest]# su guest
[guest@10 ~]$ touch readfile.c
[guest@10 ~]$ gcc readfile.c -o readfile
[guest@10 ~]$ su root
Пароль:
[root@10 guest]# chown root /home/guest/readfile.c
[root@10 guest]# chmod 700 /home/guest/readfile.c
[root@10 guest]# su guest
[guest@10 ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@10 ~]$ su root
Пароль:
[root@10 guest]# chown root:guest /home/guest/readfile
[root@10 guest]# chmod u+s /home/guest/readfile
[root@10 guest]# su guest

[guest@10 ~]$ ./readfile readfile.c
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>
#include <fcntl.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. 0.7: действия с readfile

Также попробовал прочитать файл etc/shadow: (рис. @fig:008)

```
[guest@10 ~]$ ./readfile /etc/shadow
root:$6$1AGz/j.o.lwjhwtW$nj1C3cQbGbrz8KAZx8Ti06Rpv6jTq9/Az2tkLJGD2Ev8Vwqw3QmyDxz4i65oWDYyocHfiNSzjFbWi86lL7Ccfl::0:99999:7:::
bin:!:18397:0:99999:7:::
daemon:!:18397:0:99999:7:::
adm:!:18397:0:99999:7:::
lp:!:18397:0:99999:7:::
sync:!:18397:0:99999:7:::
shutdown:!:18397:0:99999:7:::
halt:!:18397:0:99999:7:::
mail:!:18397:0:99999:7:::
operator:!:18397:0:99999:7:::
games:!:18397:0:99999:7:::
ftp:!:18397:0:99999:7:::
nobody:!:18397:0:99999:7:::
dbus:!!:18884::::::
systemd-coredump:!!:18884::::::
systemd-resolve:!!:18884::::::
tss:!!:18884::::::
polkitd:!!:18884::::::
geoclue:!!:18884::::::
rtkit:!!:18884::::::
pipewire:!!:18884::::::
pulse:!!:18884::::::
libstoragemgmt:!!:18884::::::
qemu:!!:18884::::::
usbmuxd:!!:18884::::::
unbound:!!:18884::::::
gluster:!!:18884::::::
rpc:!!:18884:0:99999:7:::
avahi:!!:18884::::::
```

Рис. 0.8: чтение etc/shadow

Оба файла прочитались несмотря на отсутствие прав у пользователя guest, поскольку программа запускается пользователем guest от имени суперпользователя.

Далее, начал работать с битом Sticky: (рис. @fig:009)

```

[guest@10 home]$ ls -l / | grep tmp
drwxrwxrwt. 11 root root 4096 ноя 11 17:48 tmp
[guest@10 home]$ echo "test" > /tmp/file01.txt
[guest@10 home]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 11 17:53 /tmp/file01.txt
[guest@10 home]$ chmod o+rw /tmp/file01.txt
[guest@10 home]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 11 17:53 /tmp/file01.txt
[guest@10 home]$ su Tim
su: пользователь Tim не существует
[guest@10 home]$ su tim
Пароль:
[tim@10 home]$ cat /tmp/file01.txt
test
[tim@10 home]$ echo "test" >> /tmp/file01.txt
[tim@10 home]$ cat /tmp/file01.txt
test
test
[tim@10 home]$ echo "test3" > /tmp/file01.txt
[tim@10 home]$ cat /tmp/file01.txt
test3
[tim@10 home]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[tim@10 home]$ su -
Пароль:
[root@10 ~]# chmod -t /tmp
[root@10 ~]# exit
выход
[tim@10 home]$ ls -l / | grep tmp
drwxrwxrwx. 11 root root 4096 ноя 11 17:56 tmp
[tim@10 home]$ cat /tmp/file01.txt
test3
[tim@10 home]$ echo "test2" >> /tmp/file01.txt
[tim@10 home]$ cat /tmp/file01.txt
test3
test2
[tim@10 home]$ echo "test3" > /tmp/file01.txt
[tim@10 home]$ cat /tmp/file01.txt
test3
[tim@10 home]$ rm /tmp/file01.txt
[tim@10 home]$ su
Пароль:
[root@10 home]# chmod +t /tmp
[root@10 home]# exit

```

Рис. 0.9: действия с Sticky-битом

Выяснил, что на директорию tmp установлен этот бит, тк среди атрибутов в конец есть t. Создал файл file01.txt в директории tmp и записал в него слово test. Посмотрел расширенные атрибуты и добавил чтение и запись для других пользователей. От пользователя tim (не явл. владельцем) прочитал файл, дозаписал в него и перезаписал его. Затем попробовал удалить файл, но безуспешно. Далее от имени root удалил атрибут t (Sticky бит) с директории tmp и повторил все действия от имени пользователя tim. На этот раз вышло то же самое, но еще удалось удалить файл. Это потому, что Sticke-бит защищает файлы, запрещая удаление файлов не владельцами. В конце вернул Sticky бит директории tmp.

## Выводы

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