

Отчёт по лабораторной работе №5

Дискреционное разграничение прав в Linux. Исследование влияния
дополнительных атрибутов

Агеева Лада Нпибд-01-19

Содержание

Цель работы	4
Выполнение лабораторной работы	5
Подготовка	5
Изучение механики SetUID	6
Исследование Sticky-бита	10
Выводы	13
Список литературы	14

Список иллюстраций

1	подготовка к работе	5
2	программа simpleid	6
3	результат программы simpleid	6
4	программа simpleid2	7
5	результат программы simpleid2	8
6	программа readfile	8
7	результат программы readfile	9
8	результат программы readfile	9
9	исследование Sticky-бита	12
10	исследование Sticky-бита	12

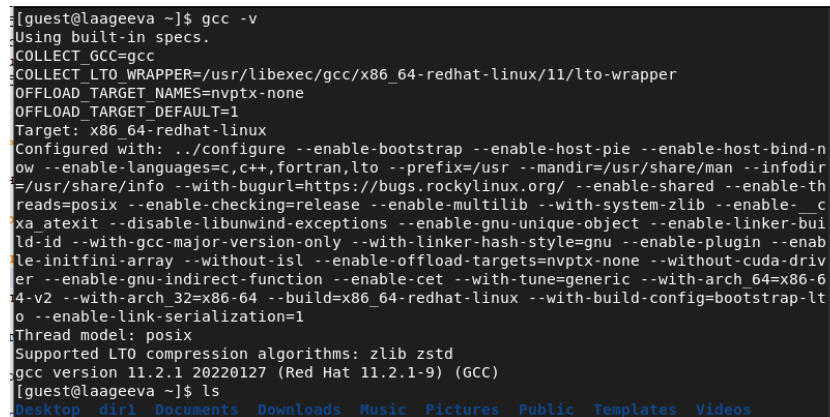
Цель работы

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

Подготовка

1. Для выполнения части заданий требуются средства разработки приложений. Проверили наличие установленного компилятора gcc командой `gcc -v`: компилятор обнаружен.
2. Чтобы система защиты SELinux не мешала выполнению заданий работы, отключили систему запретов до очередной перезагрузки системы командой `setenforce 0`:
3. Команда `getenforce` вывела `Permissive`:

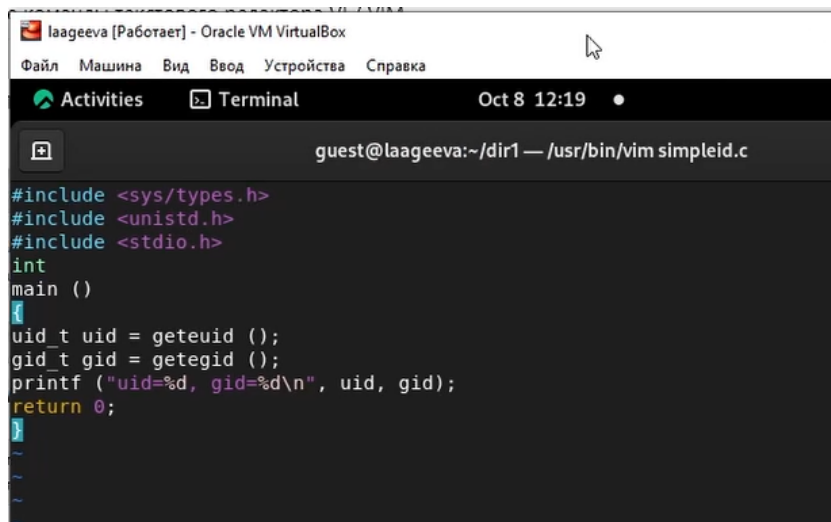


```
[guest@laageeva ~]$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-redhat-linux
Configured with: ../configure --enable-bootstrap --enable-host-pie --enable-host-bind-n
ow --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share/man --infodir
=/usr/share/info --with-bugurl=https://bugs.rockylinux.org/ --enable-shared --enable-th
reads=posix --enable-checking=release --enable-multilib --with-system-zlib --enable-_c
xa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-bui
ld-id --with-gcc-major-version-only --with-linker-hash-style=gnu --enable-plugin --enab
le-initfini-array --without-isl --enable-offload-targets=nvptx-none --without-cuda-driv
er --enable-gnu-indirect-function --enable-cet --with-tune=generic --with-arch_64=x86-6
4-v2 --with-arch_32=x86-64 --build=x86_64-redhat-linux --with-build-config=bootstrap-lt
o --enable-link-serialization=1
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.2.1 20220127 (Red Hat 11.2.1-9) (GCC)
[guest@laageeva ~]$ ls
desktop dirl Documents Downloads Music Pictures Public Templates Videos
```

Рис. 1: подготовка к работе

Изучение механики SetUID

1. Вошли в систему от имени пользователя guest.
2. Написали программу simpleid.c.



```
laageeva [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Activities  Terminal  Oct 8 12:19
guest@laageeva:~/dir1 — /usr/bin/vim simpleid.c

#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;
}
```

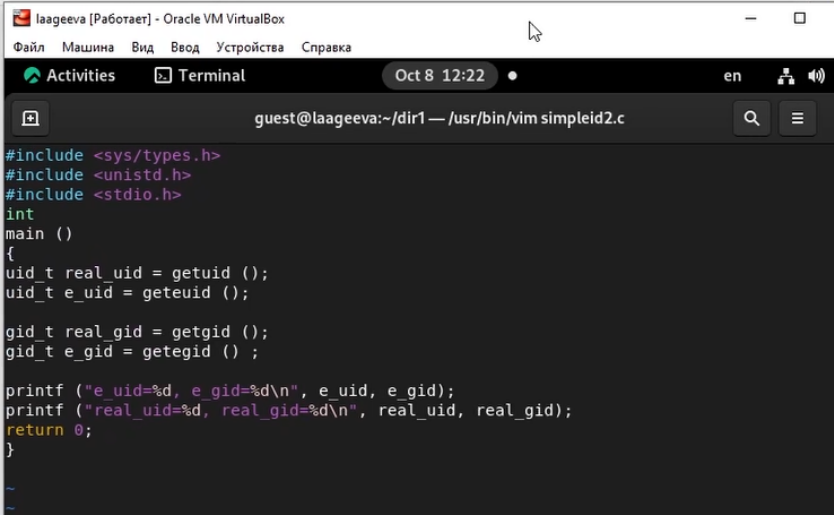
Рис. 2: программа simpleid

3. Скомпилировали программу и убедились, что файл программы создан: gcc simpleid.c -o simpleid
4. Выполнили программу simpleid командой ./simpleid
5. Выполнили системную программу id с помощью команды id. uid и gid совпадает в обеих программах

```
[guest@laageeva ~]$ cd dir1
[guest@laageeva dir1]$ touch simpleid.c
[guest@laageeva dir1]$ vi simpleid.c
[guest@laageeva dir1]$ gcc simpleid.c -o simpleid
[guest@laageeva dir1]$ ./simpleid
uid=1001, gid=1001
[guest@laageeva dir1]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 3: результат программы simpleid

6. Усложнили программу, добавив вывод действительных идентификаторов.



```
#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: программа simpleid2

7. Скомпилировали и запустили simpleid2.c:

```
gcc simpleid2.c -o simpleid2
```

```
./simpleid2
```

8. От имени суперпользователя выполнили команды:

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

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

9. Использовали su для повышения прав до суперпользователя

10. Выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

```
ls -l simpleid2
```

11. Запустили simpleid2 и id:

./simpleid2

id

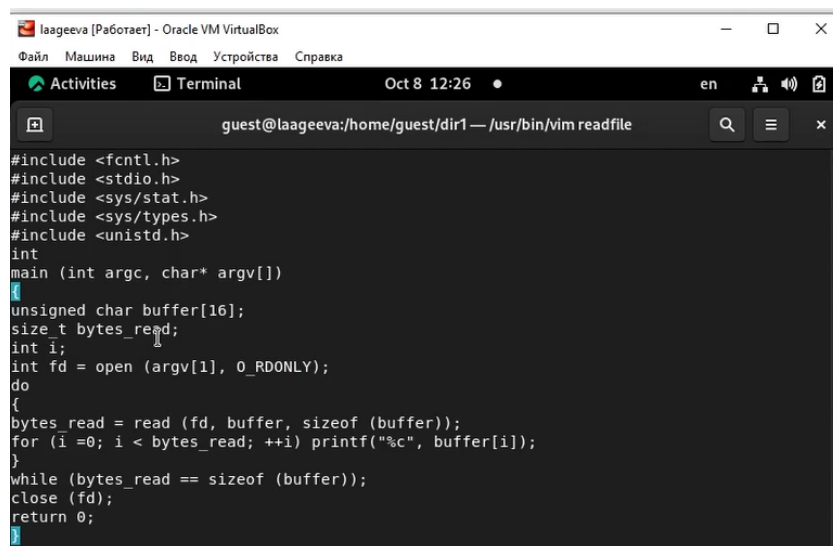
Результат выполнения программ теперь немного отличается

12. Проделали тоже самое относительно SetGID-бита.

```
[guest@laageeva dir1]$ vi simpleid2.c
[guest@laageeva dir1]$ gcc simpleid2.c -o simpleid2
[guest@laageeva dir1]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@laageeva dir1]$ su
Password:
[root@laageeva dir1]# chown root:guest /home/guest/dir1/simpleid2
[root@laageeva dir1]# chmod u+s /home/guest/dir1/simpleid2
[root@laageeva dir1]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct  8 12:22 simpleid2
[root@laageeva dir1]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@laageeva dir1]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 5: результат программы simpleid2

13. Написали 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. 6: программа readfile

14. Откомпилировали её.


```
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
```

16. Проверили, что пользователь guest не может прочитать файл readfile.c.
17. Сменили у программы readfile владельца и установили SetU'D-бит.
18. Проверили, может ли программа readfile прочитать файл readfile.c
19. Проверили, может ли программа readfile прочитать файл /etc/shadow

```
[root@laageeva dir1]# touch readfile
[root@laageeva dir1]# vi readfile
[root@laageeva dir1]# gcc readfile.c -o readfile
cc1: fatal error: readfile.c: No such file or directory
compilation terminated.
[root@laageeva dir1]# mv readfile readfile.c
[root@laageeva dir1]# gcc readfile.c -o readfile
[root@laageeva dir1]# chown root:root readfile
[root@laageeva dir1]# chmod -r readfile.c
[root@laageeva dir1]# chmod u+s readfile
[root@laageeva dir1]# exit
exit
[guest@laageeva dir1]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@laageeva 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){
```

Рис. 7: результат программы readfile

```
[guest@laageeva dir1]$ ./readfile /etc/shadow
root:$6$U3hBnGT3004LMYfc$0AVMNo.XFJ/mc5dkqog934Cap2u50PM6xHwma62P8zodqfvgPs7yakei.kzuo5
idns20vEAaHRBUvgoKFre.1::0:99999:7:::
bin:*.19123:0:99999:7:::
daemon:*.19123:0:99999:7:::
adm:*.19123:0:99999:7:::
lp:*.19123:0:99999:7:::
sync:*.19123:0:99999:7:::
shutdown:*.19123:0:99999:7:::
halt:*.19123:0:99999:7:::
mail:*.19123:0:99999:7:::
operator:*.19123:0:99999:7:::
```

Рис. 8: результат программы readfile

Исследование Sticky-бита

1. Выяснили, установлен ли атрибут Sticky на директории /tmp:

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

2. От имени пользователя guest создали файл file01.txt в директории /tmp со словом test:

```
echo "test" > /tmp/file01.txt
```

3. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные»:

```
ls -l /tmp/file01.txt
```

```
chmod o+rw /tmp/file01.txt
```

```
ls -l /tmp/file01.txt
```

Первоначально все группы имели право на чтение, а запись могли осуществлять все, кроме «остальных пользователей».

4. От пользователя (не являющегося владельцем) попробовали прочитать файл /file01.txt:

```
cat /file01.txt
```

5. От пользователя попробовали дозаписать в файл /file01.txt слово test3 командой:

```
echo "test2" >> /file01.txt
```

6. Проверили содержимое файла командой:

```
cat /file01.txt
```

В файле теперь записано:

Test

Test2

7. От пользователя попробовали записать в файл `/tmp/file01.txt` слово `test4`, стерев при этом всю имеющуюся в файле информацию командой. Для этого воспользовалась командой `echo "test3" > /tmp/file01.txt`

8. Проверили содержимое файла командой

```
cat /tmp/file01.txt
```

9. От пользователя попробовали удалить файл `/tmp/file01.txt` командой `rm /tmp/file01.txt`, однако получила отказ.

10. От суперпользователя командой выполнили команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`:

```
chmod -t /tmp
```

Покинули режим суперпользователя командой `exit`.

11. От пользователя проверили, что атрибута `t` у директории `/tmp` нет:

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

12. Повторили предыдущие шаги. Получилось удалить файл

13. Удалось удалить файл от имени пользователя, не являющегося его владельцем.

14. Повысили свои права до суперпользователя и вернули атрибут `t` на директорию `/tmp` :

```
su
```

```
chmod +t /tmp
```

```
exit
```

```

[guest@laageeva dir1]$ echo "test" > /tmp/file01.txt
[guest@laageeva dir1]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Oct  8 12:33 /tmp/file01.txt
[guest@laageeva dir1]$ chmod o+rw /tmp/file01.txt
[guest@laageeva dir1]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Oct  8 12:33 /tmp/file01.txt
[guest@laageeva dir1]$ su gueast2
su: user gueast2 does not exist or the user entry does not contain all the required fields
[guest@laageeva dir1]$ su guest2
Password:
su: Authentication failure
[guest@laageeva dir1]$ su guest2
Password:
su: Authentication failure
[guest@laageeva dir1]$ su guest2
Password:
su: Authentication failure
[guest@laageeva dir1]$ su guest2
Password:
[guest2@laageeva dir1]$ cat /tmp/file01.txt
test
[guest2@laageeva dir1]$ echo "test2" > /tmp/file01.txt
[guest2@laageeva dir1]$ cat /tmp/file01.txt
test2
[guest2@laageeva dir1]$ echo "test3" > /tmp/file01.txt
[guest2@laageeva dir1]$ cat /tmp/file01.tx
cat: /tmp/file01.tx: No such file or directory
[guest2@laageeva dir1]$ cat /tmp/file01.txt

```

Рис. 9: исследование Sticky-бита

```

[guest2@laageeva dir1]$ cat /tmp/file01.txt
test3
[guest2@laageeva dir1]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': No such file or directory
[guest2@laageeva dir1]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': No such file or directory
[guest2@laageeva dir1]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@laageeva dir1]$ su -
Password:
[root@laageeva ~]# chmod -t /tmp
[root@laageeva ~]# exit
logout
[guest2@laageeva dir1]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 Oct  8 12:38 tmp
[guest2@laageeva dir1]$ cat file01.txt
cat: file01.txt: Permission denied
[guest2@laageeva dir1]$ echo "test4" > /tmp/file01.txt
[guest2@laageeva dir1]$ echo "test5" >> /tmp/file01.txt
[guest2@laageeva dir1]$ rm file01.txt
rm: cannot remove 'file01.txt': Permission denied
[guest2@laageeva dir1]$ su -
Password:
su: Authentication failure
[guest2@laageeva dir1]$ su -
Password:
[root@laageeva ~]# chmod +t /tmp
[root@laageeva ~]# exit
logout

```

Рис. 10: исследование Sticky-бита

Выводы

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

Список литературы

1. КОМАНДА CHATTR В LINUX
2. chattr