

Отчёт по лабораторной работе №5

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

Гаглов Олег Мелорович

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 4 |
| 2 | Выполнение лабораторной работы | 5 |
| 2.1 | Подготовка | 5 |
| 2.2 | Изучение механики SetUID | 6 |
| 2.3 | Исследование Sticky-бита | 10 |
| 3 | Выводы | 14 |
| | Список литературы | 15 |

Список иллюстраций

| | | |
|------|-----------------------------------|----|
| 2.1 | проверка | 5 |
| 2.2 | simpleid | 6 |
| 2.3 | компиляция и выполнение | 6 |
| 2.4 | улучшение simpleid | 7 |
| 2.5 | id и simpleid | 7 |
| 2.6 | проверка simpleid | 8 |
| 2.7 | readfile.c | 8 |
| 2.8 | смена владельца | 9 |
| 2.9 | проверка guest | 9 |
| 2.10 | проверка readfile | 9 |
| 2.11 | проверка etc/shadow | 10 |
| 2.12 | grep tmp | 10 |
| 2.13 | запись в файл | 11 |
| 2.14 | проверка файла | 12 |
| 2.15 | Alt text | 13 |

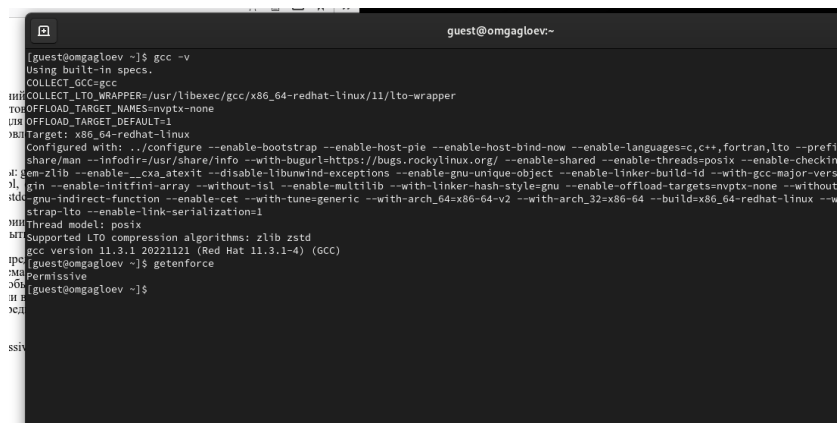
1 Цель работы

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.

2 Выполнение лабораторной работы

2.1 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@omgagloev:~$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/11/lto-wrapper
TOPOFFLOAD_TARGET_NAMES=nvptx-none
TOPOFFLOAD_TARGET_DEFAULT=1
Target: x86_64-redhat-linux
Configured with: ../configure --enable-bootstrap --enable-host-pie --enable-host-bind-now --enable-languages=c,c++,fortran,lto --prefix=/usr --infodir=/usr/share/info --with-bugurl=https://bugs.rockylinux.org/ --enable-shared --enable-threads=posix --enable-checking=release --enable-cxx-abi-compat --enable-cxa-ehc --enable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-gcc-major-version=11 --enable-initfini-array --without-isl --enable-multilib --with-linker-hash-style=gnu --enable-offload-targets=nvptx-none --without-slib --enable-gnu-indirect-function --enable-cet --with-tune=generic --with-arch_64=x86-64-v2 --with-arch_32=x86-64 --build=x86_64-redhat-linux --with-strap-lto --enable-link-serialization=1
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.3.1 20221121 (Red Hat 11.3.1-4) (GCC)

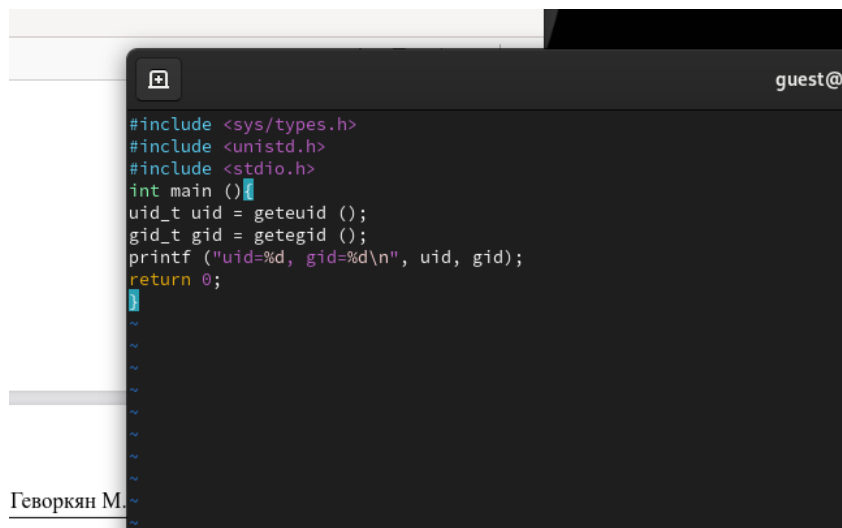
guest@omgagloev:~$ getenforce
Permissive

guest@omgagloev:~$
```

Рис. 2.1: проверка

2.2 Изучение механики SetUID

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



```
guest@
#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. 2.2: simpleid

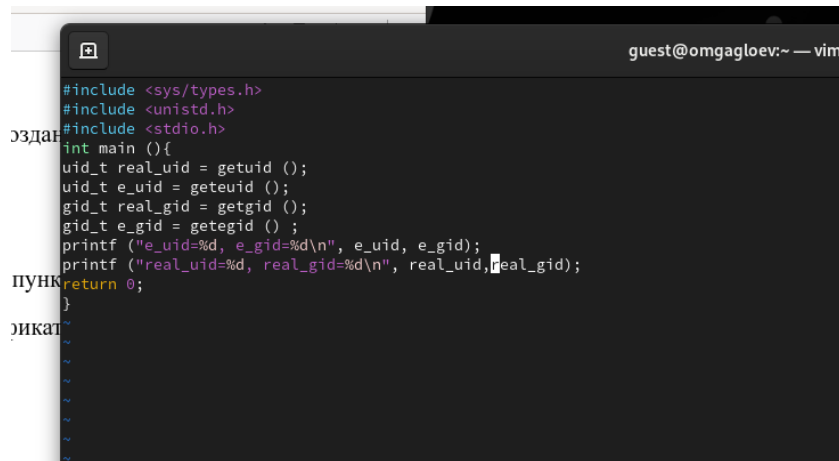
3. Скомпилировали программу и убедились, что файл программы создан: gcc simpleid.c -o simpleid
4. Выполнили программу simpleid командой ./simpleid



```
[guest@omgagloev ~]$ gcc simpleid.c -o simpleid
[guest@omgagloev ~]$ ./simpleid
uid=1001, gid=1001
[guest@omgagloev ~]$
```

Рис. 2.3: компиляция и выполнение

5. Выполнили системную программу id с помощью команды id. uid и gid совпадает в обеих программах
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. 2.4: улучшение simpleid

7. Скомпилировали и запустили simpleid2.c:



```
[guest@omgagloev ~]$ gcc simpleid.c -o simpleid
[guest@omgagloev ~]$ ./simpleid
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@omgagloev ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@omgagloev ~]$
```

Рис. 2.5: id и simpleid

8. От имени суперпользователя выполнили команды:

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

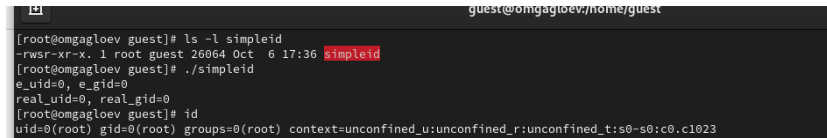
```
chmod u+s /home/guest/simpleid2
```

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:

Результат выполнения программ теперь немного отличается

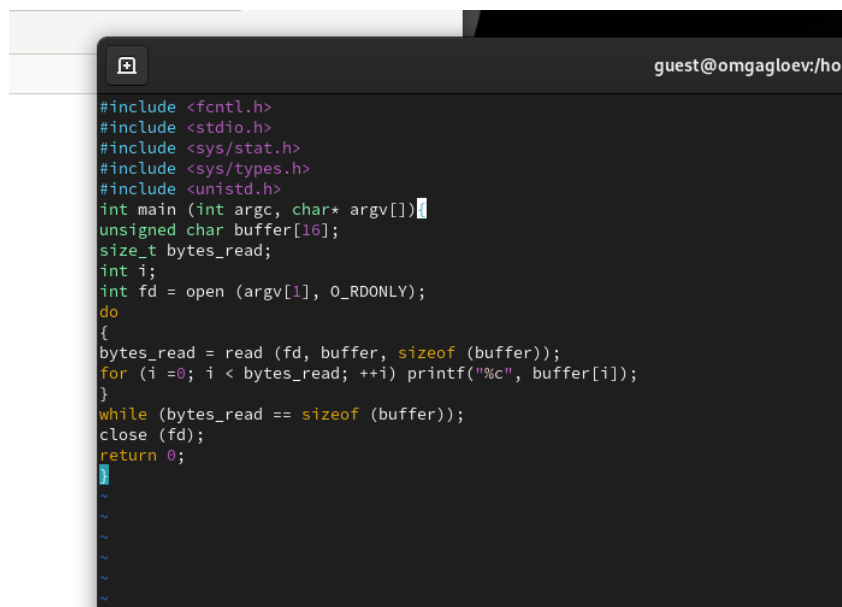


```
guest@omgagloev/home/guest
[root@omgagloev guest]# ls -l simpleid
-rwsr-xr-x. 1 root guest 26064 Oct  6 17:36 simpleid
[root@omgagloev guest]# ./simpleid
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@omgagloev guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 2.6: проверка simpleid

12. Проделали тоже самое относительно SetGID-бита.

13. Написали программу readfile.c



```
guest@omgagloev:/ho
#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: readfile.c

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
```

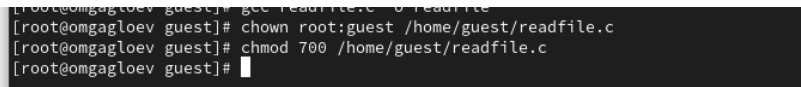


Рис. 2.8: смена владельца

16. Проверили, что пользователь `guest` не может прочитать файл `readfile.c`.

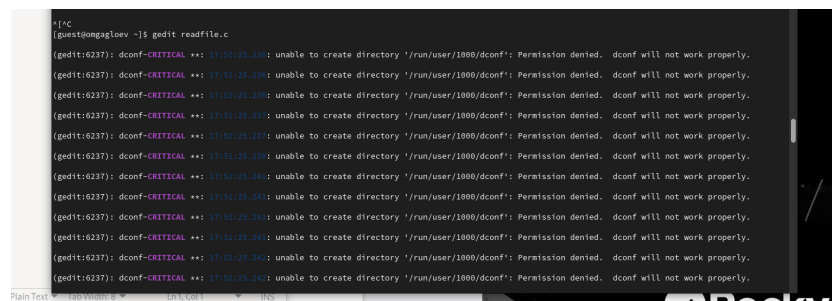


Рис. 2.9: проверка guest

17. Сменили у программы readfile владельца и установили SetU'D-бит.

18. Проверили, может ли программа readfile прочитать файл readfile.c

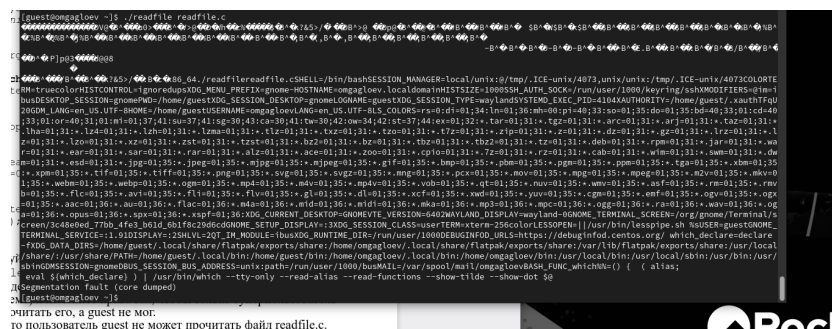


Рис. 2.10: проверка readfile

3. Проверили, может ли программа readfile прочитать файл /etc/shadow

```
[root@omgagloev guest]# ./readfile /etc/shadow
root:$6$uGfYdW/8x2FuNq$0sz1ICs49ZehieBMXnNzrY5vn9z1lBhaVuG0g60zf/DW8fKkpYnPk1JZ5sFJk1I9ZrQkvPOSbgUGZD.ivQt/::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:::
shutdown:*:19469:0:99999:7:::
root:*:19469:0:99999:7:::
mail:*:19469:0:99999:7:::
operator:*:19469:0:99999:7:::
games:*:19469:0:99999:7:::
ftp:*:19469:0:99999:7:::
nobody:*:19469:0:99999:7:::
systemd-coredump:::19609::::::
dbus:::19609::::::
polkitd:::19609::::::
avahi:::19609::::::
rtkit:::19609::::::
sssd:::19609::::::
pipewire:::19609::::::
libstoragemgmt:::19609::::::
systemd-oom:::19609::::::
```

Рис. 2.11: проверка etc/shadow

2.3 Исследование Sticky-бита

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

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

```
guest@omgagloev:/home/guest
[guest@omgagloev ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Oct  6 17:54 tmp
[guest@omgagloev ~]$ echo "test" > /tmp/file01.txt
[guest@omgagloev ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Oct  6 17:58 /tmp/file01.txt
[guest@omgagloev ~]$ chmod o+rw /tmp/file01.txt
[guest@omgagloev ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Oct  6 17:58 /tmp/file01.txt
[guest@omgagloev ~]$
```

Рис. 2.12: grep tmp

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

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

```
bash: /tmp/file01.txt: Permission denied
[guest2@omgagloev ~]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@omgagloev ~]$
```

Рис. 2.13: запись в файл

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
```

```

bash: /tmp/file01.txt: Permission denied
[guest2@omgagloev ~]$ cat /tmp/file01.txt
test
[guest2@omgagloev ~]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@omgagloev ~]$ cat /tmp/file01.txt
test
[guest2@omgagloev ~]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': No such file or directory
[guest2@omgagloev ~]$

```

Рис. 2.14: проверка файла

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
```

```

rm: cannot remove '/tmp/file01.txt': No such file or directory
[guest2@omgagloev ~]$ su -
Password:
[root@omgagloev ~]# chmod -t /tmp
[root@omgagloev ~]# exit
logout
[guest2@omgagloev ~]$ ls -l / | grep tmp
drwxrwxrwx. 19 root root 4096 Oct  6 18:03 tmp
[guest2@omgagloev ~]$

```

Покинули режим су-

перпользователя 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
```

```
5y drwxrwxrwx. 19 root root 4096 Oct 6 18:05 tmp
[root@omgagloev ~]# chmod +t /tmp
[root@omgagloev ~]# exit
logout
B chmod: changing permissions of '/tmp': Operation not permitted
exit
[root@omgagloev guest]# ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Oct 6 18:06 tmp
[root@omgagloev guest]#
```

Рис. 2.15: Alt text

3 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