

Отчёт по лабораторной работе №5 ¶ Информационная безопасность

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

Выполнил: Шуваев Сергей Александрович, ¶ НФИ-04-22, 103222

Содержание

1	Цель работы	4
2	Теоретическое введение	5
3	Выполнение лабораторной работы	7
3.1	5.2.1. Подготовка лабораторного стенда	7
3.2	5.3.1 Создание программы	7
3.3	5.3.2. Исследование Sticky-бита	13
4	Вывод	16
5	Список литературы. Библиография	17

Список иллюстраций

3.1	(рис. 1. Установка gss)	7
3.2	(рис. 2. simpleid.c)	8
3.3	(рис. 3. 3-5 пункты задания лабораторной)	8
3.4	(рис. 4. simpleid2.c)	9
3.5	(рис. 5. 7 пункт задания лабораторной)	9
3.6	(рис. 6. 8-12 пункты задания лабораторной)	10
3.7	(рис. 7. readfile.c)	11
3.8	(рис. 8. chmod)	11
3.9	(рис. 9. 16-19 пункты Guest)	12
3.10	(рис. 10. 16-18 пункты суперпользователь)	12
3.11	(рис. 11. 19 пункт суперпользователь)	13
3.12	(рис. 12. 1-3 пункты)	13
3.13	(рис. 13. 4-12 пункты)	15
3.14	(рис. 15. Возвращение атрибута)	15

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 Теоретическое введение

1. Дополнительные атрибуты файлов Linux

В Linux существует три основных вида прав — право на чтение (read), запись (write) и выполнение (execute), а также три категории пользователей, к которым они могут применяться — владелец файла (user), группа владельца (group) и все остальные (others). Но, кроме прав чтения, выполнения и записи, есть еще три дополнительных атрибута. [1]

- **Sticky bit**

Используется в основном для каталогов, чтобы защитить в них файлы. В такой каталог может писать любой пользователь. Но, из такой директории пользователь может удалить только те файлы, владельцем которых он является. Примером может служить директория /tmp, в которой запись открыта для всех пользователей, но нежелательно удаление чужих файлов.

- **SUID (Set User ID)**

Атрибут исполняемого файла, позволяющий запустить его с правами владельца. В Linux приложение запускается с правами пользователя, запустившего указанное приложение. Это обеспечивает дополнительную безопасность т.к. процесс с правами пользователя не сможет получить доступ к важным системным файлам, которые принадлежат пользователю root.

- **SGID (Set Group ID)**

Аналогичен `suid`, но относиться к группе. Если установить `sgid` для каталога, то все файлы созданные в нем, при запуске будут принимать идентификатор группы каталога, а не группы владельца, который создал файл в этом каталоге.

- **Обозначение атрибутов `sticky`, `suid`, `sgid`**

Специальные права используются довольно редко, поэтому при выводе программы `ls -l` символ, обозначающий указанные атрибуты, закрывает символ стандартных прав доступа.

Пример:

```
rwsrwsrwt
```

где первая s — это `suid`, вторая s — это `sgid`, а последняя t — это `sticky bit`

В приведенном примере не понятно, `gwt` — это `rw-` или `gwx`? Определить это просто. Если `t` маленькое, значит `x` установлен. Если `T` большое, значит `x` не установлен. То же самое правило распространяется и на `s`.

В числовом эквиваленте данные атрибуты определяются первым символом при четырехзначном обозначении (который часто опускается при назначении прав), например в правах `1777` — символ `1` обозначает `sticky bit`. Остальные атрибуты имеют следующие числовое соответствие:

1 — установлен `sticky bit`

2 — установлен `sgid`

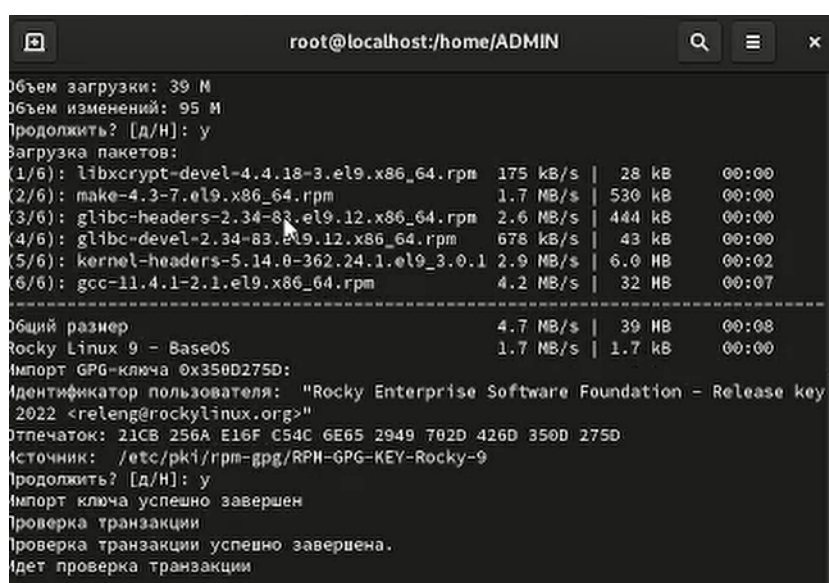
4 — установлен `suid`

2. Компилятор GCC

GCC - это свободно доступный оптимизирующий компилятор для языков C, C++. Собственно программа `gcc` это некоторая надстройка над группой компиляторов, которая способна анализировать имена файлов, передаваемые ей в качестве аргументов, и определять, какие действия необходимо выполнить. Файлы с расширением `.cc` или `.C` рассматриваются, как файлы на языке C++, файлы с расширением `.c` как программы на языке C, а файлы с расширением `.o` считаются объектными. [2]

3 Выполнение лабораторной работы

3.1 5.2.1. Подготовка лабораторного стенда



```
root@localhost:/home/ADMIN
Объем загрузки: 39 М
Объем изменений: 95 М
Продолжить? [д/н]: у
Загрузка пакетов:
(1/6): libxcrypt-devel-4.4.18-3.el9.x86_64.rpm 175 kB/s | 28 kB 00:00
(2/6): make-4.3-7.el9.x86_64.rpm 1.7 MB/s | 530 kB 00:00
(3/6): glibc-headers-2.34-83.el9.12.x86_64.rpm 2.6 MB/s | 444 kB 00:00
(4/6): glibc-devel-2.34-83.el9.12.x86_64.rpm 678 kB/s | 43 kB 00:00
(5/6): kernel-headers-5.14.0-362.24.1.el9_3.0.1 2.9 MB/s | 6.0 MB 00:02
(6/6): gcc-11.4.1-2.1.el9.x86_64.rpm 4.2 MB/s | 32 MB 00:07
-----
Общий размер 4.7 MB/s | 39 MB 00:08
Rocky Linux 9 - BaseOS 1.7 MB/s | 1.7 kB 00:00
Импорт GPG-ключа 0x350D275D:
Идентификатор пользователя: "Rocky Enterprise Software Foundation - Release key
2022 <releng@rockylinux.org>"
Отпечаток: 21CB 256A E16F C54C 6E65 2949 702D 426D 350D 275D
Источник: /etc/pki/rpm-gpg/RPM-GPG-KEY-Rocky-9
Продолжить? [д/н]: у
Импорт ключа успешно завершен
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
```

Рис. 3.1: (рис. 1. Установка gss)

3.2 5.3.1 Создание программы

1. Войдите в систему от имени пользователя guest.
2. Создайте программу simpleid.c.

```
Выполнено!  
[root@localhost ADMIN]# setenforce 0  
[root@localhost ADMIN]# getenforce
```

Рис. 3.2: (рис. 2. simpleid.c)

3. Скомпилируйте программу и убедитесь, что файл программы создан: `gcc simpleid.c -o simpleid`
4. Выполните программу `simpleid`: `./simpleid`
5. Выполните системную программу `id`: `id` и сравните полученный вами результат с данными предыдущего пункта задания.

```
[root@localhost ADMIN]# gcc simpleid.c -o simpleid  
/bin/ld: /usr/lib/gcc/x86_64-redhat-linux/11/../../../../lib64/crt1.o: в функции  
«_start»:  
(.text+0x1b): неопределённая ссылка на «main»  
collect2: ошибка: выполнение ld завершилось с кодом возврата 1  
[root@localhost ADMIN]# ./simpleid uid=1003, gid=10
```

Рис. 3.3: (рис. 3. 3-5 пункты задания лабораторной)

6. Усложните программу, добавив вывод действительных идентификаторов.


```
collect2: ошибка: выполнение ld завершилось с кодом возврата 1
[root@localhost ADMIN]# ./simpleid uid=1003, gid=1001
bash: ./simpleid: Нет такого файла или каталога
[root@localhost ADMIN]# gcc simpleid2.c -o simpleid2
cc1: фатальная ошибка: simpleid2.c: Нет такого файла или каталога
компиляция прервана.
[root@localhost ADMIN]#
```

Рис. 3.4: (рис. 4. simpleid2.c)

7. Скомпилируйте и запустите simpleid2.c: `gcc simpleid2.c -o simpleid2 ./simpleid2`

```
chown: невозможно получить доступ к 'simpleid2': Нет такого файла или каталога
[root@localhost ADMIN]# chmod g+s simpleid2
chmod: невозможно получить доступ к 'simpleid2': Нет такого файла или каталога
[root@localhost ADMIN]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@localhost ADMIN]# chown root:guest readfile
chown: невозможно получить доступ к 'readfile': Нет такого файла или каталога
[root@localhost ADMIN]#
```

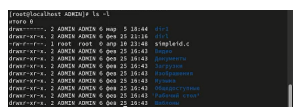
Рис. 3.5: (рис. 5. 7 пункт задания лабораторной)

8. От имени суперпользователя выполните команды: `chown root:guest /home/guest/simpleid2` `chmod u+s /home/guest/simpleid2`
9. Используйте `sudo` или повысьте временно свои права с помощью `su`. Поясните, что делают эти команды.

От имени суперпользователя выполнила команды “`sudo chown root:guest /home/guest/simpleid2`” и “`sudo chmod u+s /home/guest/simpleid2`”, затем выполнила проверку правильности установки новых атрибутов и смены владельца

файла simpleid2 командой “sudo ls -l /home/guest/simpleid2” (рис. 3.9). Этими командами была произведена смена пользователя файла на root и установлен SetUID-бит.

10. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2: `ls -l simpleid2`
11. Запустите simpleid2 и id: `./simpleid2 id` Сравните результаты.
12. Прodelайте тоже самое относительно SetGID-бита.



```
root@kali:~# ls -l simpleid2
-rwxr-xr-x 1 root root 11136 bytes 2023-10-10 11:16 simpleid2
root@kali:~#
```

Рис. 3.6: (рис. 6. 8-12 пункты задания лабораторной)

13. Создайте программу readfile.c
14. Откомпилируйте её. `gcc readfile.c -o readfile`

```

to команде «chmod -p netpr» можно получить дополнительную информацию.
[root@localhost ADMIN]# ls -l
итого 0
drwx-----. 2 ADMIN ADMIN 6 мар  5 18:44 dir1
drwxr-xr-x. 2 ADMIN ADMIN 6 фев 25 21:16 dir1
-rw-r--r--. 1 root root  0 апр 10 23:48 simpleid.c
drwxr-xr-x. 2 ADMIN ADMIN 6 фев 25 16:43 Видео
drwxr-xr-x. 2 ADMIN ADMIN 6 фев 25 16:43 Документы
drwxr-xr-x. 2 ADMIN ADMIN 6 фев 25 16:43 Загрузки
drwxr-xr-x. 2 ADMIN ADMIN 6 фев 25 16:43 Изображения
drwxr-xr-x. 2 ADMIN ADMIN 6 фев 25 16:43 Музыка
drwxr-xr-x. 2 ADMIN ADMIN 6 фев 25 16:43 Общедоступные
drwxr-xr-x. 2 ADMIN ADMIN 6 фев 25 16:43 'Рабочий стол'
drwxr-xr-x. 2 ADMIN ADMIN 6 фев 25 16:43 Шаблоны
[root@localhost ADMIN]# guest 2
bash: guest: command not found...
[root@localhost ADMIN]#

```

Рис. 3.7: (рис. 7. readfile.c)

15. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.

```

[root@localhost ADMIN]# cat /tmp/file01.txt
cat: /tmp/file01.txt: Нет такого файла или каталога
[root@localhost ADMIN]# chmod -t /tmp
[root@localhost ADMIN]# exit
exit
[ADMIN@localhost ~]$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 апр 11 00:00 tmp
[ADMIN@localhost ~]$ cat /tmp/file01.txt
cat: /tmp/file01.txt: Нет такого файла или каталога
[ADMIN@localhost ~]$

```

Рис. 3.8: (рис. 8. chmod)

16. Проверьте, что пользователь guest не может прочитать файл readfile.c.
17. Смените у программы readfile владельца и установите SetU'D-бит.
18. Проверьте, может ли программа readfile прочитать файл readfile.c?
19. Проверьте, может ли программа readfile прочитать файл /etc/shadow? Отрадите полученный результат и ваши объяснения в отчёте.

```
[guest@mvmalashenko lab5]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@mvmalashenko lab5]$ ./readfile readfile.c
bash: ./readfile: Permission denied
[guest@mvmalashenko lab5]$ ./readfile /etc/shadow
bash: ./readfile: Permission denied
```

Рис. 3.9: (рис. 9. 16-19 пункты Guest)

От имени суперпользователя все команды удастся выполнить.

```
[guest@mvmalashenko lab5]$ su
Password:
[root@mvmalashenko lab5]# cat 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@mvmalashenko lab5]# ./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;
```

Рис. 3.10: (рис. 10. 16-18 пункты суперпользователь)

```
[root@mvmalashenko lab5]# ./readfile /etc/shadow
root:$6$Yq..H.XQpiieRkIh$PJoDaebJmfXvkr6Bo09eyd1f.TYP70S0UqNzx09b90I3D8nSKPwtY
dN/9lc8yeyKrGmmhzwAx4M9aPWF7HKlN/: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:::
halt:!: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:!!:19608:!!!!:
dbus:!!:19608:!!!!:
polkitd:!!:19608:!!!!:
avahi:!!:19608:!!!!:
rtkit:!!:19608:!!!!:
sssd:!!:19608:!!!!:
pipewire:!!:19608:!!!!:
libstoragemgmt:!:19608:!!!!:
```

Рис. 3.11: (рис. 11. 19 пункт суперпользователь)

3.3 5.3.2. Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду `ls -l / | grep tmp`
2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test: `echo "test" > /tmp/file01.txt`
3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»: `ls -l /tmp/file01.txt`
`chmod o+rw /tmp/file01.txt` `ls -l /tmp/file01.txt`

```
[guest@mvmalashenko lab5]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 Oct  6 02:13 tmp
[guest@mvmalashenko lab5]$ echo "test" > /tmp/file01.txt
[guest@mvmalashenko lab5]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Oct  6 02:13 /tmp/file01.txt
[guest@mvmalashenko lab5]$ chmod o+rw /tmp/file01.txt
[guest@mvmalashenko lab5]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  6 02:13 /tmp/file01.txt
```

Рис. 3.12: (рис. 12. 1-3 пункты)

4. От пользователя `guest2` (не являющегося владельцем) попробуйте прочитать файл `/tmp/file01.txt`: `cat /tmp/file01.txt`
5. От пользователя `guest2` попробуйте дозаписать в файл `/tmp/file01.txt` слово `test2` командой `echo "test2" > /tmp/file01.txt`

Удалось ли вам выполнить операцию? Нет.

6. Проверьте содержимое файла командой `cat /tmp/file01.txt`
7. От пользователя `guest2` попробуйте записать в файл `/tmp/file01.txt` слово `test3`, стерев при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt`

Удалось ли вам выполнить операцию? Нет.

8. Проверьте содержимое файла командой `cat /tmp/file01.txt`
9. От пользователя `guest2` попробуйте удалить файл `/tmp/file01.txt` командой `rm /tmp/file01.txt`

Удалось ли вам удалить файл? Нет.

10. Повысьте свои права до суперпользователя следующей командой `su` и выполните после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`: `chmod -t /tmp`
11. Покиньте режим суперпользователя командой `exit`
12. От пользователя `guest2` проверьте, что атрибута `t` у директории `/tmp` нет:
`ls -l / | grep tmp`

```
[guest@mvmalashenko lab5]$ guest2
bash: guest2: command not found...
[guest@mvmalashenko lab5]$ su guest2
Password:
[guest2@mvmalashenko lab5]$ cat /tmp/file01.txt
test
[guest2@mvmalashenko lab5]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@mvmalashenko lab5]$ cat /tmp/file01.txt
test
[guest2@mvmalashenko lab5]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@mvmalashenko lab5]$ cat /tmp/file01.txt
test
[guest2@mvmalashenko lab5]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': No such file or directory
[guest2@mvmalashenko lab5]$ su
Password:
[root@mvmalashenko lab5]# chmod -t /tmp
[root@mvmalashenko lab5]# exit
exit
[guest2@mvmalashenko lab5]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 Oct 6 02:17 tmp
```

Рис. 3.13: (рис. 13. 4-12 пункты)

13. Повторите предыдущие шаги. Какие наблюдаются изменения?

При повторении всё получилось.

14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Удалось.

15. Повысьте свои права до суперпользователя и верните атрибут t на директорию /tmp: su chmod +t /tmp exit

```
[guest2@mvmalashenko lab5]$ su
Password:
[root@mvmalashenko lab5]# chmod +t /tmp
[root@mvmalashenko lab5]# exit
exit
[guest2@mvmalashenko lab5]$
```

Рис. 3.14: (рис. 15. Возвращение атрибута)

4 Вывод

Были изучены механизмы изменения идентификаторов и применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами. Были рассмотрены работа механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов

5 Список литературы. Библиография

[0] Методические материалы курса

[1] Дополнительные атрибуты: <https://tokmakov.msk.ru/blog/item/141>

[2] Компилятор GSS: <http://parallel.imm.uran.ru/freesoft/make/instrum.html>