

Отчёт по лабораторной работе №5

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

Даниил Анатольевич Вейценфельд

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	SetUID, SetGID	8
4.2	Sticky-бит	12
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Создание программы	8
4.2	Код программы получения ID	8
4.3	Запуск от guest	9
4.4	Усложненный код получения ID	9
4.5	Установка владельца и SUID-бита	9
4.6	Проверка от пользователя guest	9
4.7	Установка SGID-бита и проверка	10
4.8	Программа для чтения	10
4.9	Создан текстовый файл	11
4.10	Проверка чтения программой	11
4.11	Чтение /etc/shadow	12
4.12	Sticky-бит установлен на д-и /tmp	12
4.13	Создание файла file01	12
4.14	Эксперименты от guest2 с файлом	13
4.15	Удаление sticky-бита	13
4.16	Успешное удаление файла	13

Список таблиц

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 Задание

1. Создать программу для проверки uid и gid
2. Протестировать ее от имени разных пользователей
3. Протестировать ее с разными правами доступа и SUID/SGID-битами
4. Создать программу для чтения файла
5. Протестировать ее на файлах с разными правами доступа и SUID/SGID-битами
6. Протестировать sticky-бит

3 Теоретическое введение

SetUID- и SetGID-биты (они же SUID, SGID) являются “заменой” обычным битам на право выполнения (x) файла; имеют обозначения s, через `ls -l` отображаются на месте бита x пользователя или группы соответственно.

Эти биты позволяют выполнить файл любому пользователю от лица владельца файла. Для примера: программа `sudo` имеет владельца `root` и SUID бит, значит любой пользователь может ее выполнить от лица `root`.

Sticky Bit - если он установлен для папки, то файлы в этой папке могут быть удалены только их владельцем. Пример использования этого бита в операционной системе это системная папка `/tmp`. Эта папка разрешена на запись любому пользователю, но удалять файлы в ней могут только пользователи, являющиеся владельцами этих файлов.

Теорию работы в *nix-системах см. в [1–6].

4 Выполнение лабораторной работы

4.1 SetUID, SetGID

С помощью языка Си и компилятора gcc создана (рис. 4.1) простая программа для проверки uid и gid исполняющего пользователя (рис. 4.2).

```
(base) mkdir src
(base) cd src/
(base) nano simpleid.c
(base) gcc simpleid.c -o simpleid
(base) ./simpleid
```

Рис. 4.1: Создание программы

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int main() {
    uid_t uid = geteuid();
    uid_t gid = getegid();
    printf("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

Рис. 4.2: Код программы получения ID

Программа верно определила id пользователя guest (рис. 4.3). Команда id возвращает ту же информацию.


```
[guest@weizenfeld src]$ ./simpleid
uid=1001, gid=1001
[guest@weizenfeld src]$
```

Рис. 4.3: Запуск от guest

Немного усложним программу для получения как отображаемых, так и реальных ID (рис. 4.4).

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int main() {
    uid_t uid = geteuid();
    uid_t gid = getegid();

    uid_t uid_r = getuid();
    uid_t gid_r = getgid();

    printf("uid=%d, gid=%d\n", uid, gid);
    printf("real ones: uid=%d, gid=%d\n", uid_r, gid_r);
    return 0;
}
```

Рис. 4.4: Усложненный код получения ID

Установим ей владельца root и SUID-бит (рис. 4.5).

```
(base) [root@weizenfeld src]# chown root:guest simpleid2
(base) [root@weizenfeld src]# chmod u+s simpleid2
(base) [root@weizenfeld src]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 18256 окт 8 19:27 simpleid2
```

Рис. 4.5: Установка владельца и SUID-бита

Проверим, что она выводит для пользователя guest: рис. 4.6.

```
[guest@weizenfeld src]$ ./simpleid2
uid=0, gid=1001
real ones: uid=1001, gid=1001
```

Рис. 4.6: Проверка от пользователя guest

Установим ей и SGID-бит вместе с группой root. Информация верна. (suid-бит был ранее снят) (рис. 4.7)

```

(base) sudo chmod g+s simpleid2
(base) ls -l simpleid2
-rwxrwsr-x. 1 root root 18256 окт  8 19:27 simpleid2
(base) su user
su: пользователь user не существует
(base) su guest
Пароль:
[guest@weizenfeld src]$ simpleid2
bash: simpleid2: команда не найдена...
[guest@weizenfeld src]$ ./simpleid2
uid=1001, gid=0
real ones: uid=1001, gid=1001
[guest@weizenfeld src]$

```

Рис. 4.7: Установка SGID-бита и проверка

Теперь напишем программу, которая читает файл как cat (рис. 4.8).

```

#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

int main(int argc, char* argv[]) {
    if (argc < 2) {
        printf("Too few arguments");
        return 1;
    }

    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. 4.8: Программа для чтения

Создадим текстовый файл с правами на чтение и запись только для root (рис. 4.9).

```
(base) echo "Hello, world!" > test.txt
(base) sudo chown root:root test.txt
(base) sudo chmod a-r test.txt
(base) ls -l test.txt
--w--w----. 1 root root 14 окт  8 19:43 test.txt
(base) sudo chmod ug+r test.txt
(base) ls -l test.txt
-rw-rw----. 1 root root 14 окт  8 19:43 test.txt
```

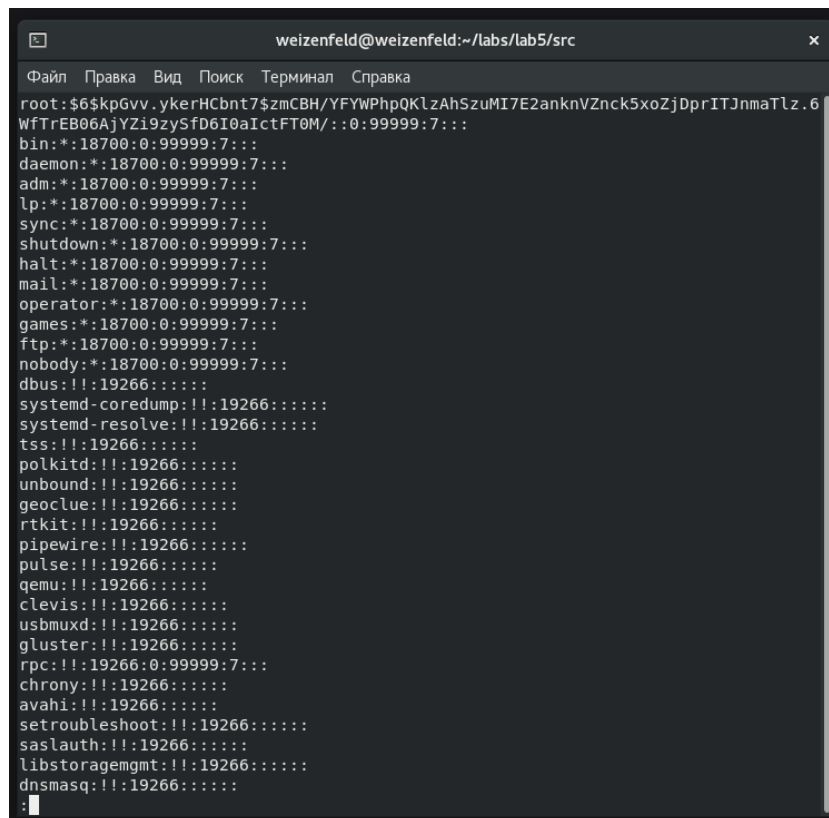
Рис. 4.9: Создан текстовый файл

Проверим чтение с помощью `cat` и с помощью программы (рис. 4.10). Программе ранее был установлен бит SUID.

```
(base) cat test.txt
cat: test.txt: Permission denied
(base) ./readfile test.txt
Hello, world!
```

Рис. 4.10: Проверка чтения программой

Также, удалось считать файл `/etc/shadow` с помощью `./readfile /etc/shadow | less` (рис. 4.11).

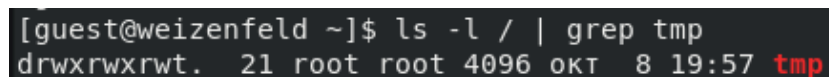


```
weizenfeld@weizenfeld:~/labs/lab5/src
Файл Правка Вид Поиск Терминал Справка
root:$6$kpGvv.ykerHCbnt7$zmCBH/YFYWPhpQKlZAhSzuMI7E2anknVZnck5xoZjDprITJnmaTlz.6
WfTrEB06AjYZi9zySfd6I0aIctFT0M/::0:99999:7:::
bin:!:18700:0:99999:7:::
daemon:!:18700:0:99999:7:::
adm:!:18700:0:99999:7:::
lp:!:18700:0:99999:7:::
sync:!:18700:0:99999:7:::
shutdown:!:18700:0:99999:7:::
halt:!:18700:0:99999:7:::
mail:!:18700:0:99999:7:::
operator:!:18700:0:99999:7:::
games:!:18700:0:99999:7:::
ftp:!:18700:0:99999:7:::
nobody:!:18700:0:99999:7:::
dbus:!!:19266::::::
systemd-coredump:!!:19266::::::
systemd-resolve:!!:19266::::::
tss:!!:19266::::::
polkitd:!!:19266::::::
unbound:!!:19266::::::
geoclue:!!:19266::::::
rtkit:!!:19266::::::
pipewire:!!:19266::::::
pulse:!!:19266::::::
qemu:!!:19266::::::
clevis:!!:19266::::::
usbmuxd:!!:19266::::::
gluster:!!:19266::::::
rpc:!!:19266:0:99999:7:::
chrony:!!:19266::::::
avahi:!!:19266::::::
setroubleshoot:!!:19266::::::
saslauth:!!:19266::::::
libstoragemgmt:!!:19266::::::
dnsmasq:!!:19266::::::
:
```

Рис. 4.11: Чтение /etc/shadow

4.2 Sticky-бит

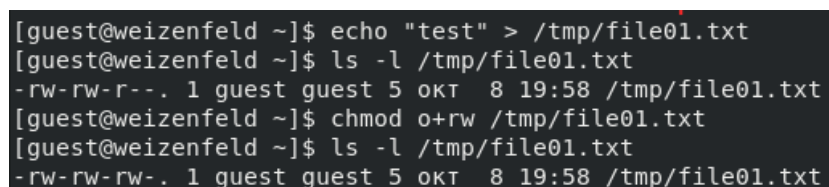
Проверим его наличие на директории /tmp (рис. 4.12).



```
[guest@weizenfeld ~]$ ls -l / | grep tmp
drwxrwxrwt. 21 root root 4096 окт 8 19:57 tmp
```

Рис. 4.12: Sticky-бит установлен на д-и /tmp

Создадим там файл с правами на чтение/запись для всех (рис. 4.13).



```
[guest@weizenfeld ~]$ echo "test" > /tmp/file01.txt
[guest@weizenfeld ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт 8 19:58 /tmp/file01.txt
[guest@weizenfeld ~]$ chmod o+rw /tmp/file01.txt
[guest@weizenfeld ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 окт 8 19:58 /tmp/file01.txt
```

Рис. 4.13: Создание файла file01

Теперь попробуем чтение, запись и дозапись от другого пользователя. Все успешно. Но удалить файл нельзя (рис. 4.14).

```
[guest@weizenfeld ~]$ su guest2
Пароль:
[guest2@weizenfeld guest]$ cat /tmp/file01.txt
test
[guest2@weizenfeld guest]$ echo "222" >> /tmp/file01.txt
[guest2@weizenfeld guest]$ cat /tmp/file01.txt
test
222
[guest2@weizenfeld guest]$ echo "332" > /tmp/file01.txt
[guest2@weizenfeld guest]$ cat /tmp/file01.txt
332
[guest2@weizenfeld guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Operation not permitted
[guest2@weizenfeld guest]$
```

Рис. 4.14: Эксперименты от guest2 с файлом

Уберем от пользователя root sticky-бит с директории /tmp (рис. 4.15).

```
[guest2@weizenfeld guest]$ su -
Пароль:
(base) [root@weizenfeld ~]# chmod -t /tmp
(base) [root@weizenfeld ~]# logout
[guest2@weizenfeld guest]$ ls -l / | grep tmp
drwxrwxrwx. 20 root root 4096 окт  8 20:02 tmp
```

Рис. 4.15: Удаление sticky-бита

Файл так же можно читать/записывать, но теперь можно и удалить (рис. 4.16).

```
[guest2@weizenfeld guest]$ echo "32" > /tmp/file01.txt
[guest2@weizenfeld guest]$ cat /tmp/file01.txt
32
[guest2@weizenfeld guest]$ rm /tmp/file01.txt
```

Рис. 4.16: Успешное удаление файла

5 Выводы

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

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.