

Лабораторная работа №5

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

Ильин Андрей Владимирович

Содержание

1	Цель работы	4
2	Задачи	5
3	Теоретическое введение	6
3.1	Термины	6
3.2	Окружение	6
4	Выполнение лабораторной работы	8
4.1	Создание программы	8
4.2	Исследование Sticky-бита	13
5	Анализ результатов	16
6	Выводы	17
	Список литературы	18

Список иллюстраций

4.1	Запуск <code>simpleid.c</code> и <code>simpleid2.c</code>	9
4.2	Запуск <code>simpleid2.c</code> с измененным владельцем и атрибутами . .	10
4.3	Подготовка к запуску <code>readfile</code>	12
4.4	Запуск <code>readfile (readfile.c)</code>	12
4.5	Запуск <code>readfile (/etc/shadow)</code>	13
4.6	Атрибут <code>Sticky</code>	14
4.7	Исследование <code>Sticky (1)</code>	14
4.8	Исследование <code>Sticky (2)</code>	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 Задачи

1. Создать программу способную выводить `gid`, `uid` и провести исследование SetUID-битов.
2. Исследовать Sticky-бит.

3 Теоретическое введение

3.1 Термины

- Терминал (или «Bash», сокращение от «Bourne-Again shell») — это программа, которая используется для взаимодействия с командной оболочкой. Терминал применяется для выполнения административных задач, например: установку пакетов, действия с файлами и управление пользователями. [1]
- Права доступа определяют, какие действия конкретный пользователь может или не может совершать с определенными файлами и каталогами. [2]
- Расширенные атрибуты файловых объектов (далее - расширенные атрибуты) - поддерживаемая некоторыми файловыми системами возможность ассоциировать с файловыми объектами произвольные метаданные. [3]
- Sticky Bit - в случае, если этот бит установлен для папки, то файлы в этой папке могут быть удалены только их владельцем. [4]

3.2 Окружение

- Rocky Linux - это корпоративная операционная система с открытым исходным кодом, разработанная таким образом, чтобы быть на 100% совместимой с Red Hat Enterprise Linux. Он находится в стадии интенсивной разработки сообществом. [5]

- Git - это распределенное программное обеспечение для контроля версиями. [6]
- VirtualBox - это кросс-платформенное ПО для виртуализации x86 и AMD64/Intel64 с открытым кодом для корпоративного и домашнего использования. [7]

4 Выполнение лабораторной работы

4.1 Создание программы

1. Создадим, скомпилируем и запустим программу `simpleid.c`. Сравним с выводом команды `id`. (рис. 4.1)

```
gcc simpleid.c -o simpleid
```

```
./simpleid
```

```
id
```

```
#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;  
}
```

2. Создадим, скомпилируем и запустим программу `simpleid2.c` (усложненная версия `simpleid.c`). (рис. 4.1)

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

```
./simpleid2
```



```

#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;
}

```



```

[guest@aviljin lab051]$ gcc simpleid.c -o simpleid
[guest@aviljin lab051]$ ./simpleid
uid=1001, gid=1001
[guest@aviljin lab051]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0
-s0:c0.c1023
[guest@aviljin lab051]$ gcc simpleid2.c -o simpleid2
[guest@aviljin lab051]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001

```

Рис. 4.1: Запуск simpleid.c и simpleid2.c

- От имени суперпользователя сменим пользователя и изменим атрибуты на simpleid2. Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2. После чего запустим simpleid2 (от имени guest) (рис. 4.2)

```

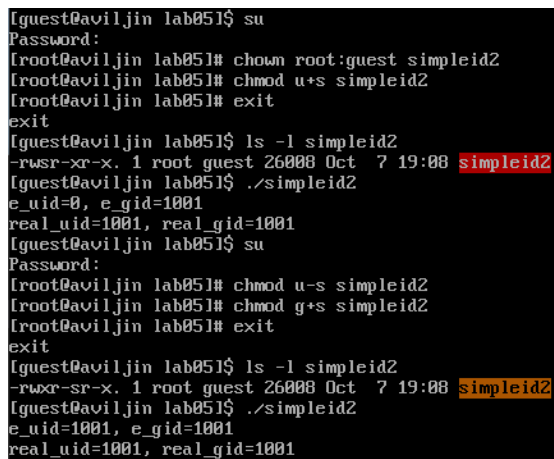
su
chown root:guest /home/guest/simpleid2
chmod u+s /home/guest/simpleid2
exit

```

```
ls -l simpleid2
./simpleid2
```

4. От имени суперпользователя установим SetGID-бит на simpleid2. Выполним проверку правильности установки новых атрибутов файла simpleid2. После чего запустим simpleid2 (от имени guest) (рис. 4.2)

```
su
chmod u-s /home/guest/simpleid2
chmod g+s /home/guest/simpleid2
exit
ls -l simpleid2
./simpleid2
```



```
[guest@aviljin lab051]$ su
Password:
[root@aviljin lab051]# chown root:guest simpleid2
[root@aviljin lab051]# chmod u-s simpleid2
[root@aviljin lab051]# exit
exit
[guest@aviljin lab051]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 26008 Oct  7 19:08 simpleid2
[guest@aviljin lab051]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aviljin lab051]$ su
Password:
[root@aviljin lab051]# chmod u-s simpleid2
[root@aviljin lab051]# chmod g+s simpleid2
[root@aviljin lab051]# exit
exit
[guest@aviljin lab051]$ ls -l simpleid2
-rwxr-sr-x. 1 root guest 26008 Oct  7 19:08 simpleid2
[guest@aviljin lab051]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Рис. 4.2: Запуск simpleid2.c с измененным владельцем и атрибутами

5. Создадим программу readfile.c. Откомпилируем ее. Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его. Проверим, что пользователь guest не может прочитать файл readfile.c. После этого сменим у программы readfile владельца и установим SetUID-бит. (рис. 4.3)

```

#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;
}

```

```

su
chown root:guest readfile.c
chmod 700 readfile.c
exit
cat readfile.c
su
chown root:guest readfile
chmod u+s readfile

```

exit

```
[guest@aviljin lab05]$ su
Password:
[root@aviljin lab05]# chown root:guest readfile.c
[root@aviljin lab05]# chmod 700 readfile.c
[root@aviljin lab05]# ls -l readfile.c
-rwx-----. 1 root guest 455 Oct  7 19:07 readfile.c
[root@aviljin lab05]# exit
exit
[guest@aviljin lab05]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@aviljin lab05]$ su
Password:
[root@aviljin lab05]# chown root:guest readfile
[root@aviljin lab05]# chmod u+s readfile
[root@aviljin lab05]# ls -l readfile
-rwsr-xr-x. 1 root guest 25952 Oct  7 19:40 readfile
```

Рис. 4.3: Подготовка к запуску readfile

6. Проверим, может ли программа readfile прочитать файлы readfile.c и /etc/shadow. (рис. 4.4, 4.5)

./readfile readfile.c

```
[guest@aviljin lab05]$ ./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;
    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;
}[guest@aviljin lab05]$ S
```

Рис. 4.4: Запуск readfile (readfile.c)

./readfile /etc/shadow

```

f1guest@aviljin lab051$ ./readfile /etc/shadow
root:$6$ty/.S/YL1K1GzpUF$AbzG39i9.6SkS1Q9h13Gum08dUh0AvzWBN7S96J4uud0bBt4F4Kf 1cB9i0BKGPd43sb.PtuHMj6
/4Geu3Ax.c0::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:!!!!:
tss:!!:19608:!!!!:
sssd:!!:19608:!!!!:
sshd:!!:19608:!!!!:
chrony:!!:19608:!!!!:
systemd-oom:!:19608:!!!!:
aviljin:$6$4sCrU82P1w44pEDa$0jeQaFthE6.jRBkymkWhs/wQi1MqP.5e8AEaeMbfGIRp.0aE2Yu91Rb8BSHnyKV1.jQKkSrIP2
sE11.7r ifwwwk0:19608:0:99999:7:::
guest:$6$CisUY0y0G5Y1tmja$FQs3U6rwjabTh0hqdGLyX1W9fgt2T.hMpX0cCAHBp1lw11FMBXq0CbHq4SK4Ak5mKtsWSFN3fc
K/zuUh5CK0o/:19616:0:99999:7:::
guest2:$6$29g46Z0N3X4NxrU$N2ef ihsg0my3bC4z5oGUqmm5o0yHZJ2.ppSsbb8G1h8NZrtkFnAJU1.jtAH39/XfKtwImKcNqs
SHKFnA0p.jMB10:19623:0:99999:7:::

```

Рис. 4.5: Запуск readfile (/etc/shadow)

4.2 Исследование Sticky-бита

1. Выясним, установлен ли атрибут Sticky на директории /tmp. От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные». (рис. 4.6)

```

ls -l / | grep tmp
echo "test" > /tmp/file01.txt
ls -l /tmp/file01.txt
chmod o+rw /tmp/file01.txt
ls -l /tmp/file01.txt

```

```

lguest@aviljin ~]$ ls -l / | grep tmp
drwxrwxrwt.  5 root root 4096 Oct  7 19:59 tmp
lguest@aviljin ~]$ echo "test" > /tmp/file01.txt
lguest@aviljin ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Oct  7 20:06 /tmp/file01.txt
lguest@aviljin ~]$ chmod o+rw /tmp/file01.txt
lguest@aviljin ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  7 20:06 /tmp/file01.txt
lguest@aviljin ~]$ _

```

Рис. 4.6: Атрибут Sticky

2. От имени пользователя guest2 проведем исследование атрибута Sticky. (рис. 4.7)

```

cat /tmp/file01.txt
echo "test2" >> /tmp/file01.txt
cat /tmp/file01.txt
echo "test3" > /tmp/file01.txt
cat /tmp/file01.txt
rm /tmp/file01.txt

```

```

lguest2@aviljin ~]$ cat /tmp/file01.txt
test
lguest2@aviljin ~]$ echo "test2" >> /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
lguest2@aviljin ~]$ cat /tmp/file01.txt
test
lguest2@aviljin ~]$ echo "test3" > /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
lguest2@aviljin ~]$ cat /tmp/file01.txt
test
lguest2@aviljin ~]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
lguest2@aviljin ~]$ _

```

Рис. 4.7: Исследование Sticky (1)

3. Удалим атрибут Sticky на директории /tmp и повторим действия из предыдущего пункта. После вернем атрибут. (рис. 4.8)

```

su -
chmod -t /tmp
cat /tmp/file01.txt

```

```

exit

ls -l / | grep tmp

echo "test2" >> /tmp/file01.txt
cat /tmp/file01.txt
echo "test3" > /tmp/file01.txt
cat /tmp/file01.txt
rm /tmp/file01.txt

su -

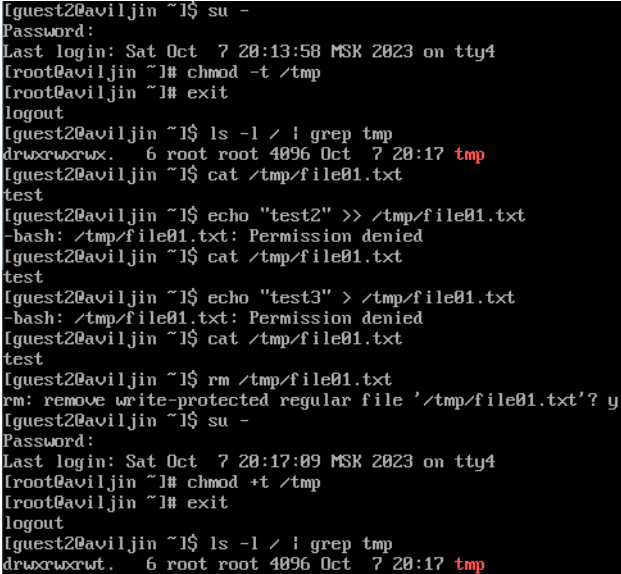
chmod +t /tmp

cat /tmp/file01.txt

exit

ls -l / | grep tmp

```



```

lguest2@aviljin ~]$ su -
Password:
Last login: Sat Oct  7 20:13:58 MSK 2023 on tty4
[root@aviljin ~]# chmod +t /tmp
[root@aviljin ~]# exit
logout
lguest2@aviljin ~]$ ls -l / | grep tmp
drwxrwxrwt.  6 root root 4096 Oct  7 20:17 tmp
lguest2@aviljin ~]$ cat /tmp/file01.txt
test
lguest2@aviljin ~]$ echo "test2" >> /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
lguest2@aviljin ~]$ cat /tmp/file01.txt
test
lguest2@aviljin ~]$ echo "test3" > /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
lguest2@aviljin ~]$ cat /tmp/file01.txt
test
lguest2@aviljin ~]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
lguest2@aviljin ~]$ su -
Password:
Last login: Sat Oct  7 20:17:09 MSK 2023 on tty4
[root@aviljin ~]# chmod +t /tmp
[root@aviljin ~]# exit
logout
lguest2@aviljin ~]$ ls -l / | grep tmp
drwxrwxrwt.  6 root root 4096 Oct  7 20:17 tmp

```

Рис. 4.8: Исследование Sticky (2)

5 Анализ результатов

Работа выполнена без непредвиденных проблем в соответствии с руководством. Ошибок и сбоев не произошло.

6 Выводы

Изучены идентификаторы SetUID-биты и Sticky-биты. Опробовали их действие на практике. Изучили влияние бита Sticky. Повысили свои навыки использования интерфейса командой строки (CLI).

Список литературы

1. Терминал Linux [Электронный ресурс]. URL: [https://www.reg.ru/blog/linux-shpargalka-komandy-terminala-dlya-novichkov/#:~:text=%D0%A2%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB%20\(%D0%B8%D0%BB%D0%B8%20%C2%AB%D0%B8%20%C2%BB%2C,%D1%81%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0%D0%BC%D0%B8%20%D0%B8%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F%D0%BC%D0%B8%7D](https://www.reg.ru/blog/linux-shpargalka-komandy-terminala-dlya-novichkov/#:~:text=%D0%A2%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB%20(%D0%B8%D0%BB%D0%B8%20%C2%AB%D0%B8%20%C2%BB%2C,%D1%81%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0%D0%BC%D0%B8%20%D0%B8%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F%D0%BC%D0%B8%7D).
2. Права доступа [Электронный ресурс]. URL: <https://codechick.io/tutorials/unix-linux/unix-linux-permissions>.
3. Расширенные атрибуты [Электронный ресурс]. URL: <https://wiki.astralin.ru/pages/viewpage.action?pageId=149063848#:~:text=Common%20Editio n%202.12-,%D0%A7%D1%82%D0%BE%20%D1%82%D0%B0%D0%BA%D0%BE%D0%B5%20%D1%80%D0%B0%D1%81%D1%88%D0%B8%D1%80%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5%20%D0%B0%D1%82%D1%80%D0%B8%D0%B1%D1%83%D1%82%D1%8B,%D1%81%20%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D1%8B%D0%BC%D0%B8%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%B0%D0%BC%D0%B8%20%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5%20%D0%BC%D0%B5%D1%82%D0%B0%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5%7D>.

4. Использование SETUID, SETGID и Sticky bit для расширенной настройки прав доступа в операционных системах Linux [Электронный ресурс]. 2021. URL: <https://ruvds.com/ru/helpcenter/suid-sgid-sticky-bit-linux/>.
5. Документация Rocky Linux [Электронный ресурс]. URL: <https://docs.rockylinux.org/>.
6. Git-Guides [Электронный ресурс]. URL: <https://github.com/git-guides>.
7. VirtualBox [Электронный ресурс]. URL: <https://www.virtualbox.org/>.