Лабораторная работа № 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Кекишева Анастасия Дмитриевна

Содержание

Список иллюстраций

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. Выполнить последовательность дейсвий, указанных в лабораторной работе [1], создавая программы и работая с битами (SetUID, SetGID, Sticky-бит), чтобы изучить влияние дополнительных атрибутов.

3 Теоретическое введение

Рассмотрим некоторые команды, которые пригодяться нам в данной лабораторной.

- chown [ПАРАМЕТР]... [ВЛАДЕЛЕЦ][:[ГРУППА]] ФАЙЛ... Эта команда позволяет сменить владельца и группу указанного ФАЙЛА на ВЛАДЕЛЬЦА и/или ГРУППУ [2].
- gcc [ИМЯ_ФАЙЛА].c -o [ИМЯ_ПРОГРАММЫ] Это команда поможет нам конвертировать файлы [1].

Рассмотрим биты, с которыми мы будем работать.

Setuid – это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо. Классический пример использования этого бита в операционной системе это команда sudo [3].

> root@ruvds-hrc [~]# which sudo /usr/bin/sudo root@ruvds-hrc [~]# ls -l /usr/bin/sudo -rwsr-xr-x 1 root root 125308 Feb 20 14:15 /usr/bin/sudo

Как мы видим на месте, где обычно установлен классический бит x (на исполнение), у нас выставлен специальный бит s. Это позволяет обычному пользователю системы выполнять команды с повышенными привилегиями без необходимости входа в систему как root, разумеется зная пароль пользователя root. Установка бита setuid не представляет сложности. Для этого используется команда:

root@ruvds-hrc [~]# chmod u+s

Аналогично setuid, бит setgid выставляется с помощью команды chmod g + s.

-rwxr-sr-x 1 root root 125308 Feb 20 14:15 /usr/bin/sudo

Последний специальный бит разрешения – это Sticky Bit . В случае, если этот бит 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 [3].

root@ruvds-hrc [~]# ls -ld /tmp drwxrwxrwt 8 root root 4096 Mar 25 10:22 /tmp

Символ «t» указывает, что на папку установлен Sticky Bit.

4 Выполнение лабораторной работы

4.1 Изучение механики SetUID

1. Вошла в систему от имени пользователя guest1. И создайте программу simpleid.c, прежде создав файл, затем записав в него код программы (рис. 4.1).

```
<http://bugzilla.redhat.com/bugzilla>.
[adkekisheva@adkekisheva ~]$ su guest1
Password:
[guest1@adkekisheva adkekisheva]$ cd
[guest1@adkekisheva ~]$ ls
Desktop  dir1  Documents  Downloads  Music  Pictures  Public  Templates  Videos
[guest1@adkekisheva ~]$ touch simpleid.c
[guest1@adkekisheva ~]$ cat >> 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;
}
^C
[guest1@adkekisheva ~]$ cat 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. 4.1: Создание программы simpleid.c

2. Скомпилировала программу командой gcc simpleid2.c -o simpleid2 и запустила simpleid2.c (рис. 4.2).

```
[guest1@adkekisheva ~]$ gcc simpleid.c -o simpleid
[guest1@adkekisheva ~]$ ./simpleid
uid=1002, gid=1002
[guest1@adkekisheva ~]$ id
uid=1002(guest1) gid=1002(guest1) groups=1002(guest1) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 4.2: Компиляция и запуск программы

3. Выполнила системную программу id. Сравнивая результаты выполнения команды ./simpleid и id, можно сказать что программа вывела только групповое и личное id, в id добавилость ещё одно групповое id (рис. 4.2).

4. Усложнила программу, добавив вывод действительных идентификаторов (рис. 4.3).
   Создав программу simpleid2.c, простотрела её, командой cat искомпилировала её (рис. 4.4).

```
[guest1@adkekisheva ~]$ touch simpleid2.c
[guest1@adkekisheva ~]$ cat >> simpleid2.c
#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;
}
^C
```

Рис. 4.3: Написание программы simpleid2.c

```
[guest1@adkekisheva ~]$ cat simpleid2.c
#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;
}
[guest1@adkekisheva ~]$ gcc simpleid2.c -o simpleid2
[guest1@adkekisheva ~]$ ./simpleid2
e_uid=1002, e_gid=1002
real_uid=1002, real_gid=1002
```

Рис. 4.4: Компиляция и запуск программы simpleid2

5. От имени суперпользователя выполнила команды:

- chown root:guest1 /home/guest1/simpleid2, которая поменяла владельца программы на root и сделала так, что этот файл принадлежит группе guest1.
- chmod u+s /home/guest/simpleid2 назначила права доступа, которые значат, что пользователь выполняет файл с разрешениями владельца файла (рис. 4.5).

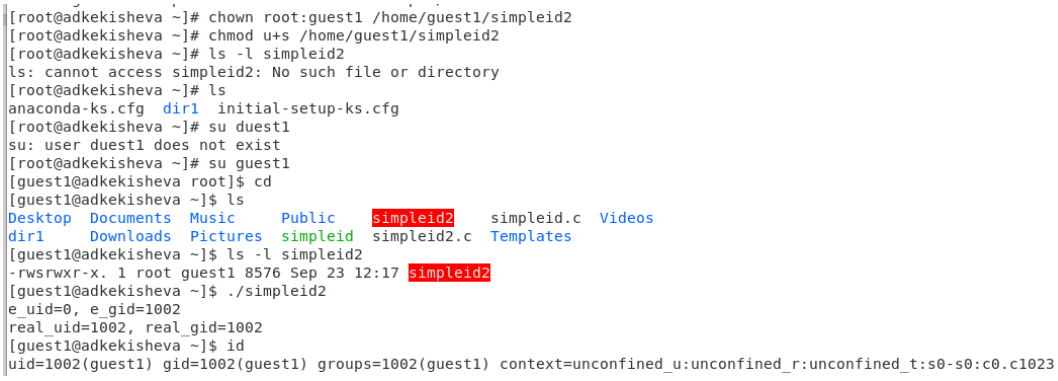

Рис. 4.5: Компиляция и запуск программы

6. Далее выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой ls -l simpleid2 и запустила simpleid2 и id: Результаты сравнения: помимо индетифткаторов юзера и группы, программа вывела нам также текущие индетификаторы (рис. 4.5).

7. Проделала тоже самое относительно SetGID-бита – для этого выполнила команду с атрибутом g: chmod g+s /home/guest/simpleid2 (рис. 4.6).

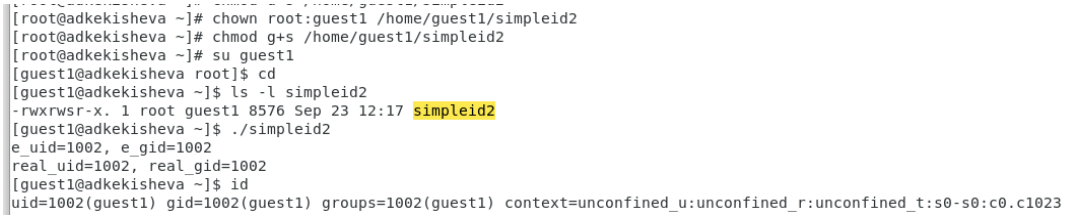

Рис. 4.6: Компиляция и запуск программы

8. Создала программу readfile.c и откомпиилировала её (рис. 4.7).

```
[guest1@adkekisheva ~]$ 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;
}
[guest1@adkekisheva ~]$ gcc readfile.c -o readfile
[guest1@adkekisheva ~]$ ls -l readfile
-rwxrwxr-x. 1 guest1 guest1 8512 Sep 23 13:13 readfile
```

Рис. 4.7: Создание и компиляция программы readfile.c

9. Смените владельца у файла readfile.c на root, настроила также и группу root, измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest1 и другие не могли. Смените у программы readfile владельца и установите SetU'D-бит. Проверила, что пользователь guest1 не может прочитать файл readfile.c. А также проверила, может ли программа readfile прочитать файл readfile.c – да может (рис. 4.8).

```
[root@adkekisheva guest1]# chown root:root readfile
[root@adkekisheva guest1]# chmod o-r readfile.c
[root@adkekisheva guest1]# chmod g-rw readfile.c
[root@adkekisheva guest1]# chmod u+s readfile
[root@adkekisheva guest1]# exit
logout
[guest2@adkekisheva ~]$ su guest1
Password:
[guest1@adkekisheva guest2]$ cd
[guest1@adkekisheva ~]$ cat readfile.c
cat: readfile.c: Permission denied
[guest1@adkekisheva ~]$ ./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;
}
```

Рис. 4.8: Настройка прав для файла readfile.c

10. Проверила, что программа readfile может прочитать файл /etc/shadow (рис. 4.9).

```
[guest1@adkekisheva ~]$ ./readfile /etc/shadow
root:$6$Y/THRikj/CG6mSx8$g91mJJwqJMCzQ1q/atgY1hjOXdKNMCc2XRdSs5WnxmGL9GVBy1Q2nSD9i9bLXLfQ8SDL4vyi/u7.oAL
ky0::0:99999:7:::
bin:*:18353:0:99999:7:::
daemon:*:18353:0:99999:7:::
adm:*:18353:0:99999:7:::
lp:*:18353:0:99999:7:::
sync:*:18353:0:99999:7:::
shutdown:*:18353:0:99999:7:::
halt:*:18353:0:99999:7:::
mail:*:18353:0:99999:7:::
operator:*:18353:0:99999:7:::
games:*:18353:0:99999:7:::
ftp:*:18353:0:99999:7:::
nobody:*:18353:0:99999:7:::
systemd-network:!!:19605::::::
dbus:!!:19605::::::
polkitd:!!:19605::::::
libstoragemgmt:!!:19605::::::
colord:!!:19605::::::
rpc:!!:19605:0:99999:7:::
saned:!!:19605::::::
saslauth:!!:19605::::::
abrt:!!:19605::::::
setroubleshoot:!!:19605::::::
rtkit:!!:19605::::::
pulse:!!:19605::::::
radvd:!!:19605::::::
chrony:!!:19605::::::
unbound:!!:19605::::::
qemu:!!:19605::::::
tss:!!:19605::::::
usbmuxd:!!:19605::::::
geoclue:!!:19605::::::
gluster:!!:19605::::::
gdm:!!:19605::::::
rpcuser:!!:19605::::::
```

Рис. 4.9: Чтение файла /etc/shadow с помощью программы readfile

4.2 Исследование Sticky-бита

1. Выяснила, установлен ли атрибут Sticky на директории /tmp, для чего выполнила команду ls -l / | grep tmp (рис. 4.10).

2. От имени пользователя guest1 создала файл file01.txt в директории /tmp со словом test: echo “test” > /tmp/file01.txt (рис. 4.10).

```
[guest1@adkekisheva ~]$ ls -l / | grep tmp
drwxrwxrwt.  27 root root 4096 Sep 24 15:16 tmp
[guest1@adkekisheva ~]$ echo "test" > /tmp/file01.txt
```

Рис. 4.10: Проверка атрибута sticky и создание файла

3. Просмотрела атрибуты (рис. 4.11) у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные»:

- ls -l /tmp/file01.txt
- chmod o+rw /tmp/file01.txt
- ls -l /tmp/file01.txt

```
[guest1@adkekisheva ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest1 guest1 5 Sep 24 15:21 /tmp/file01.txt
[guest1@adkekisheva ~]$ chmod o+rw /tmp/file01.txt
[guest1@adkekisheva ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest1 guest1 5 Sep 24 15:21 /tmp/file01.txt
[guest1@adkekisheva ~]$ su guest2
Password:
[guest2@adkekisheva guest1]$ cd
[guest2@adkekisheva ~]$ cat /tmp/file01.txt
test
```

Рис. 4.11: Добавление прав остальным пользователям на чтение и запись

4. От пользователя guest2 (не являющегося владельцем) попробовала прочитать файл /tmp/file01.txt и дозаписать в него. Дозаписать удалось, но там остался лишь новый текст, старого test не было (рис. 4.12).

```
test
[guest2@adkekisheva ~]$ echo "test2" > /tmp/file01.txt
[guest2@adkekisheva ~]$ cat /tmp/file01.txt
test2
[guest2@adkekisheva ~]$ echo "test3" > /tmp/file01.txt
[guest2@adkekisheva ~]$ cat /tmp/file01.txt
test3
[guest2@adkekisheva ~]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@adkekisheva ~]$ su -
Password:
Last login: Sun Sep 24 13:36:48 MSK 2023 on pts/0
[root@adkekisheva ~]# chmod -t /tmp
[root@adkekisheva ~]# exit
logout
[guest2@adkekisheva ~]$ ls -l / | grep tmp
drwxrwxrwx.  27 root root 4096 Sep 24 15:25 tmp
```

Рис. 4.12: Проверка атрибута sticky и создание файла

7. От пользователя guest2 попробовала записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo “test3” > /tmp/file01.txt – операция была выполнена успешно (рис. 4.12).

8. Проверьте содержимое файла и от пользователя guest2 попробовала удалить файл – файл удалить не удалось, нет прав на это. Далее повысила свои права до суперпользователя следующей командой su - и сняла атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp. Покинула режим суперпользователя командой exit. От пользователя guest2 проверила, что атрибута t у директории /tmp нет: ls -l / | grep tmp (рис. 4.13).

9. Повторите предыдущие шаги – в этот раз смогла удалить файл (рис. 4.13). После выполнения снова довавила sticky-бит.

```
[guest1@adkekisheva guest2]$ cd
[guest1@adkekisheva ~]$ echo "test" > /tmp/file01.txt
[guest1@adkekisheva ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest1 guest1 5 Sep 24 15:54 /tmp/file01.txt
[guest1@adkekisheva ~]$ su guest2
Password:
[guest2@adkekisheva guest1]$ cd
[guest2@adkekisheva ~]$ echo "test2" > /tmo/file01.txt
bash: /tmo/file01.txt: No such file or directory
[guest2@adkekisheva ~]$ echo "test2" > /tmp/file01.txt
[guest2@adkekisheva ~]$ cat /tmp/file01.txt
test2
[guest2@adkekisheva ~]$ echo "test3" > /tmp/file01.txt
[guest2@adkekisheva ~]$ cat /tmp/file01.txt
test3
[guest2@adkekisheva ~]$ rm /tmp/file01.txt
[guest2@adkekisheva ~]$ su -
Password:
Last login: Sun Sep 24 15:24:52 MSK 2023 on pts/0
[root@adkekisheva ~]# ls -l / | grep tmp
drwxrwxrwx.  28 root root 4096 Sep 24 15:56 tmp
[root@adkekisheva ~]# chmod +t /tmp
[root@adkekisheva ~]# ls -l / | grep tmp
drwxrwxrwt.  27 root root 4096 Sep 24 15:56 tmp
[root@adkekisheva ~]# exit
logout
```

Рис. 4.13: Шаги без sticky-бита

5 Выводы

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

Список литературы

1. Лабораторная работа № 5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов [Электронный ресурс]. URL: https://esystem.rudn.ru/pluginfile.php/2090417/mod_resource/content/2/005-lab_discret_sticky.pdf.
2. Команда chown Linux [Электронный ресурс]. URL: https://losst.pro/komanda-chown-linux.
3. Использование SETUID, SETGID и Sticky bit для расширенной настройки прав доступа в операционных системах Linux [Электронный ресурс]. URL: https://ruvds.com/ru/helpcenter/suid-sgid-sticky-bit-linux/.