Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Гудиева Мадина Куйраевна

Содержание

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.

Теоретические сведения

Теоретические сведения
В операционных системах Linux используются 3 базовых права доступа – на чтение (read), запись (write) и исполнение (execute). Соответственно, права назначаются пользователю (user), группе (group) и всем остальным (world). [1]

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. На месте, где обычно установлен классический бит x (на исполнение), 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. Для установки используется команда "chmod u+s ". [1]

Принцип работы Setgid очень похож на setuid с отличием, что файл будет запускаться пользователем от имени группы, которая владеет файлом. Аналогично setuid, бит setgid выставляется с помощью команды chmod g + s. Удалить эти биты можно также командой chmod, только вместо « + » используется « – » . [1]

Третий из специальных разрешений — sticky bit. Это разрешение полезно для защиты файлов от случайного удаления в среде, где несколько пользователей имеют права на запись в один и тот же каталог. Если применяется закрепленный sticky bit, пользователь может удалить файл, только если он является пользователем-владельцем файла или каталога, в котором содержится файл. По этой причине он применяется в качестве разрешения по умолчанию для каталога /tmp и может быть полезен также для каталогов общих групп. [2]

При использовании ls -ld, вы можете видеть sticky bit как t в позиции, где вы обычно видите разрешение на выполнение для других. Для sticky bit используйте chmod +t, а затем имя файла или каталога, для которого вы хотите установить разрешения. [2]

Выполнение лабораторной работы

Вошла в систему от имени пользователя guest. Создала программу simpleid.c.

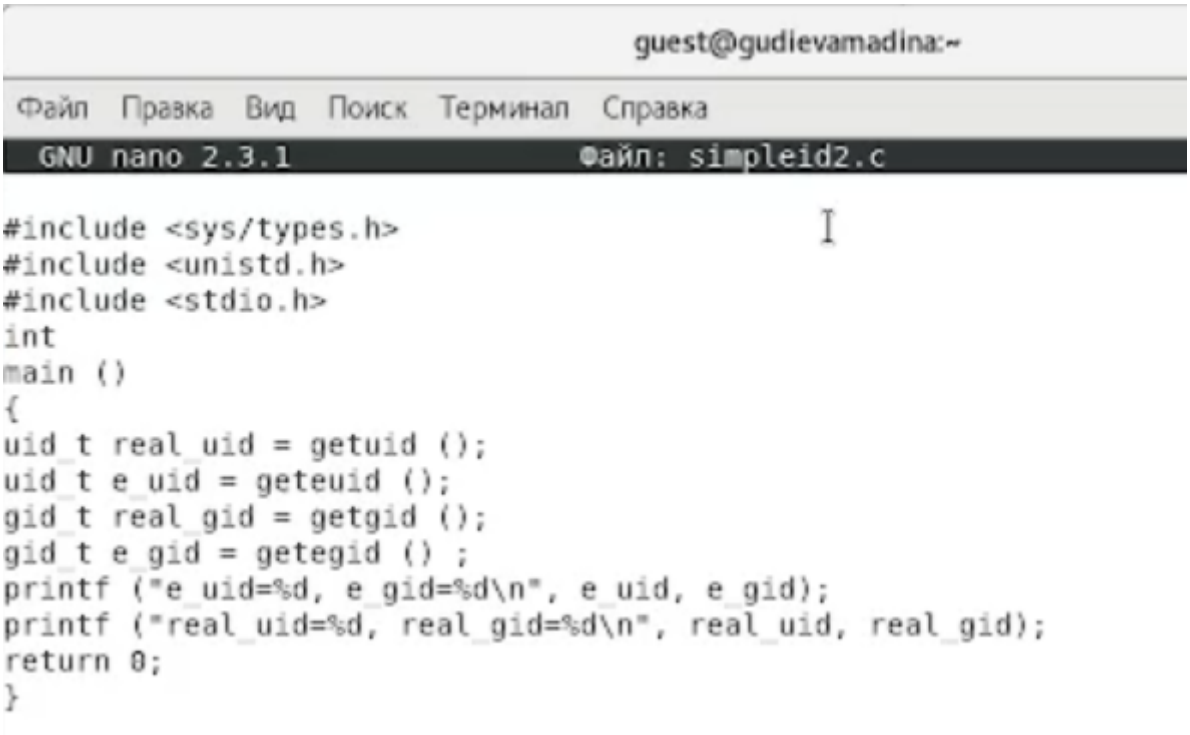
```
[guest@gudievamadina ~]$ nano simpleid.c
[guest@gudievamadina ~]$ cat simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

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

Скомпилировала и выполнила программу. Выполнив системную программу id убедилась в правильности выведенных данных

```
[guest@gudievamadina ~]$ gcc simpleid.c -o simpleid
[guest@gudievamadina ~]$ ./simpleid
uid=1001, gid=1001
[guest@gudievamadina ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u
nconfined_t:s0-s0:c0.c1023
```

Усложнила программу, добавив вывод действительных идентификаторов, назвала её simpleid2.c

```
[guest@gudievamadina ~]$ cp simpleid.c simpleid2.c
[guest@gudievamadina ~]$ nano simpleid2.c
```



Скомпилировала и запустила программу

```
[guest@gudievamadina ~]$ cp simpleid.c simpleid2.c
[guest@gudievamadina ~]$ nano simpleid2.c
```

От имени суперпользователя установила новые атрибуты и сменила владельца файла simpleid2

```
[guest@gudievamadina ~]$ su
Пароль:
[root@gudievamadina guest]# chown root:guest /home/guest/simpleid2
[root@gudievamadina guest]# chmod u+s /home/guest/simpleid2
```

Выполнила проверку и запустила программу

```
root@gudievamadina guest]# ls -l simpleid2
rwsrwxr-x. 1 root guest 8576 окт  6 21:03 simpleid2
root@gudievamadina guest]# ./simpleid2
_uid=0, e_gid=0
eal_uid=0, real_gid=0
root@gudievamadina guest]# id
id=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:
9-s0:c0.c1023
```

Проделала тоже самое относительно SetGID-бита

```
[root@gudievamadina guest]# chmod g+s /home/guest/simpleid2
```

```
[guest@gudievamadina ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 8576 окт  6 21:03 simpleid2
[guest@gudievamadina ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@gudievamadina ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:u
nconfined_t:s0-s0:c0.c1023
```

Создала программу readfile.c

```
 GNU nano 2.3.1                Файл: 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;
}
```

Откомпилировала её

```
[guest@gudievamadina ~]$ nano readfile.c
[guest@gudievamadina ~]$ gcc readfile.c -o readfile
```

Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог

```
[root@gudievamadina guest]# chown root:guest /home/guest/readfile.c
[root@gudievamadina guest]# chmod 700 /home/guest/readfile.c
[root@gudievamadina guest]# ls -l /home/guest/readfile.c
-rwx------. 1 root guest 402 окт  6 21:15 /home/guest/readfile.c
```

Убедилась, что guest не может прочитать файл readfile.c

```
[guest@gudievamadina ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
```

Смените у программы readfile владельца и установите SetU'D-бит

Убедилась, что readfile может прочитать файлы readfile.c и "/etc/shadow"

```
[root@gudievamadina guest]# chown root:guest /home/guest/readfile
[root@gudievamadina guest]# chmod u+s /home/guest/readfile
[root@gudievamadina guest]# ls -l /home/guest/readfile
-rwxrwxr-x. 1 root guest 8512 окт  6 21:15 /home/guest/readfile
```

Убедилась, что атрибут Sticky установлен на директории "/tmp"

```
[guest@gudievamadina ~]$ ./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@gudievamadina ~]$ ./readfile /etc/shadow
root:$6$0nB1ASsI9/rPsTcL$GbaS0.SSxpLdQ6PlOF74E9OKcnWDNg1F
gCyn8d2100xuNBDzh62ex8G1::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:::
```

От имени пользователя guest создала файл file01.txt в директории "/tmp" со словом test.

Разрешила чтение и запись для категории пользователей «все остальные»

```
[guest@gudievamadina ~]$ ls -l / | grep tmp
drwxrwxrwt.  20 root root 4096 окт  6 21:35 tmp
```

От пользователя guest2 просмотрела файл, успешно дозаписала и переписала его. Но не смогла его удалить

```
[guest@gudievamadina ~]$ echo "test" > /tmp/file01.txt
[guest@gudievamadina ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт  6 21:49 /tmp/file01.txt
[guest@gudievamadina ~]$ chmod o+rw /tmp/file01.txt
[guest@gudievamadina ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 окт  6 21:49 /tmp/file01.txt
[guest@gudievamadina ~]$
```

От суперпользователя сняла атрибут t (Sticky-бит) с директории "/tmp"

```
[guest@gudievamadina ~]$ su guest2
Пароль:
[guest2@gudievamadina guest]$ cat /tmp/file01.txt
test
[guest2@gudievamadina guest]$ echo "test2" > /tmp/file01.txt
[guest2@gudievamadina guest]$ cat /tmp/file01.txt
test2
[guest2@gudievamadina guest]$ echo "test3" >> /tmp/file01.txt
[guest2@gudievamadina guest]$ cat /tmp/file01.txt
test2
test3
[guest2@gudievamadina guest]$ echo "test3" > /tmp/file01.txt
[guest2@gudievamadina guest]$ cat /tmp/file01.txt
test3
[guest2@gudievamadina guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
```

Убедилась в правильности снятия атрибута и повторила предыдущие шаги. В этот раз удаление также прошло успешно

От суперпользователя вернула атрибут t (Sticky-бит) на директорию "/tmp"

```
[guest2@gudievamadina guest]$ su
Пароль:
[root@gudievamadina guest]# chmod -t /tmp
[root@gudievamadina guest]# exit
exit
[guest2@gudievamadina guest]$ ls -l / | grep tmp
drwxrwxrwx.  20 root root 4096 окт  6 21:56 tmp
[guest2@gudievamadina guest]$ cat /tmp/file01.txt
test3
[guest2@gudievamadina guest]$ echo "test3" >> /tmp/file01.txt
[guest2@gudievamadina guest]$ cat /tmp/file01.txt
test3
test3
[guest2@gudievamadina guest]$ echo "test3" > /tmp/file01.txt
[guest2@gudievamadina guest]$ cat /tmp/file01.txt
test3
[guest2@gudievamadina guest]$ rm /tmp/file01.txt
[guest2@gudievamadina guest]$ su
Пароль:
[root@gudievamadina guest]# chmod +t /tmp
[root@gudievamadina guest]# exit
exit
```

Выводы

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

Список литературы

Использование SETUID, SETGID и Sticky bit. // ruvds.com 2021. URL: https://ruvds.com/ru/helpcenter/suid-sgid-sticky-bit-linux/ (дата обращения 13.11.2021).

ИПрава в Linux (chown, chmod, SUID, GUID, sticky bit, ACL, umask). // habr.com 2019. URL: https://habr.com/ru/post/469667/ (дата обращения 13.11.2021).

Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян. Информационная безопасность компьютерных сетей: лабораторные работы. // Факультет физико-математических и естественных наук. М.: РУДН, 2015. 64 с..