Лабораторная работа №5: Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

дисциплина: Информационная безопасность

Голова Варвара Алексеевна

2021, 13 November

Цель работы

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.

Выполнение работы

Создание программы

Создала программу 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;
}
```

Figure 1: Simpleid.c

Компиляция

Скомплилировала программу и убедилась, что файл программы создан

Figure 2: Компиляция

Выполнение программы

Выполнила программу simpleid

Figure 3: Simpleid

Программа id

Выполнила системную программу id

```
uid=1003(guest_1) gid=1003(guest_1) группы=1003(guest_1) контекст=unconfined_u:u
nconfined_r:unconfined_t:s0-s0:c0.c1023
```

Figure 4: Id

Программа

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

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

int
main ()
{
uid_t real_uid = geteuid ();
gid_t e_uid = getegid ();

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;
}
```

Figure 5: Simpleid2.c

Компиляция

Скомпилировала simpleid2.c

```
[guest_1@vagolova ~]$ gcc simpleid2.c -o simpleid2
```

Figure 6: Компиляция

Суперпользователь

От имени суперпользователя выполнила команды

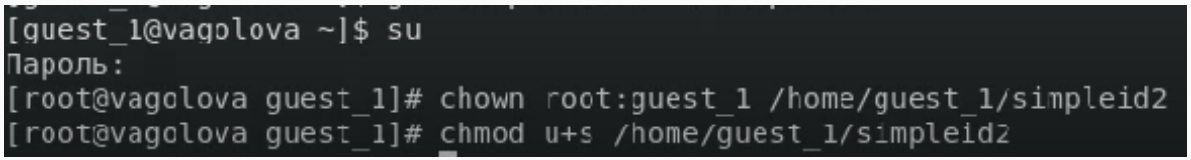


Figure 7: Команды

Проверка

Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

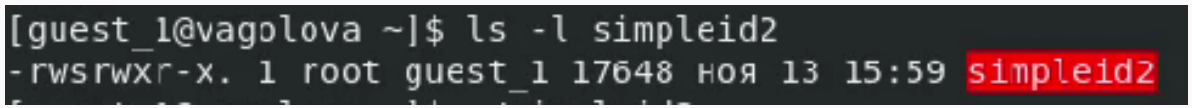


Figure 8: Проверка

Программы

Запустила simpleid2 и id

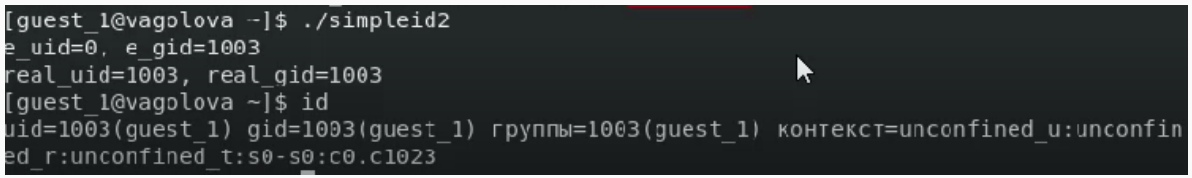


Figure 9: Запуск программ

Программа

Создала программу readfile.c

```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;
}
```

Figure 10: Readfile.c

Права readfile.c

Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь мог прочитать его, а guest_1 не мог. Проверила, что пользователь guest_1 не может прочитать файл readfile.c. Сменила у программы readfile владельца и установите SetU'D-бит.

```
[guest_1@vagolova ~]$ chmod 000 /home/guest_1/readfile.c
[guest_1@vagolova ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest_1@vagolova ~]$ chown root:root /home/guest_1/readfile.c
chown: изменение владельца '/home/guest_1/readfile.c': Операция не позволена
[guest_1@vagolova ~]$ sudo chown root:root /home/guest_1/readfile.c
[sudo] пароль для guest_1:
Попробуйте ещё раз.
[sudo] пароль для guest_1:
guest_1 is not in the sudoers file.  This incident will be reported.
[guest_1@vagolova ~]$ su
Пароль:
[root@vagolova guest_1]# chown root:root /home/guest_1/simpleid2
[root@vagolova guest_1]# chown root:guest_1 /home/guest_1/readfile
[root@vagolova guest_1]# chmod u+s /home/guest_1/readfile
```

Figure 11: Права

Атрибуты

Выяснила, установлен ли атрибут Sticky на директории /tmp. От имени пользователя guest_1 создала файл file01.txt в директории /tmp со словом test. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные»

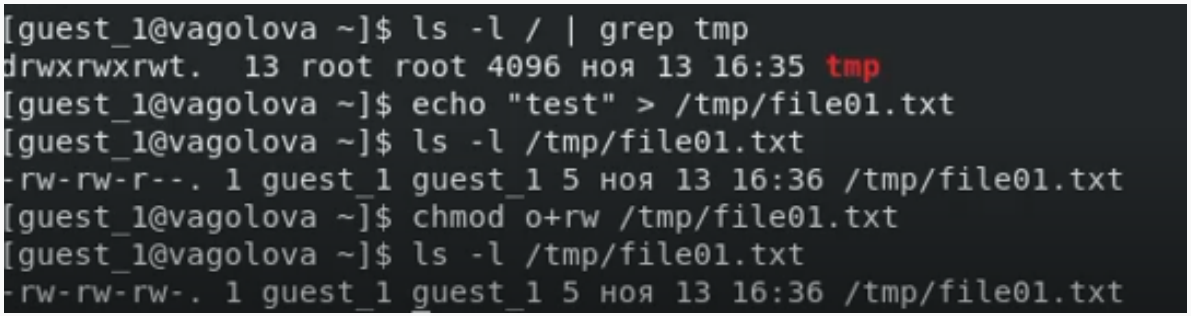
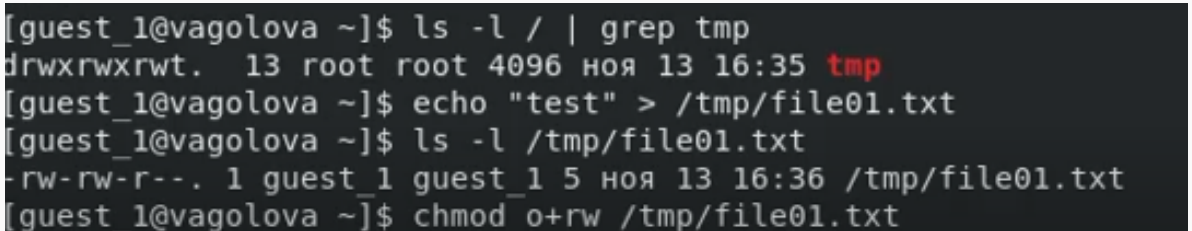


Figure 12: Атрибуты

File01

От пользователя guest_2 попробовала прочитать файл /tmp/file01.txt. От пользователя guest_2 попробовала дозаписать в файл /tmp/file01.txt слово test2. Проверила содержимое файла командой. От пользователя guest_2 попробовала записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию. Проверила содержимое файла командой. От пользователя guest_2 попробовала удалить файл /tmp/file01.txt. Повысила свои права до суперпользователя и выполнила после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp. Покинула режим суперпользователя. От пользователя guest_2 проверила, что атрибута t у директории /tmp нет.



Атрибут t

Повторила предыдущие шаги. Мне удалось удалить файл от имени пользователя, не являющегося его владельцем. Повысила свои права до суперпользователя и вернула атрибут t на директорию /tmp.

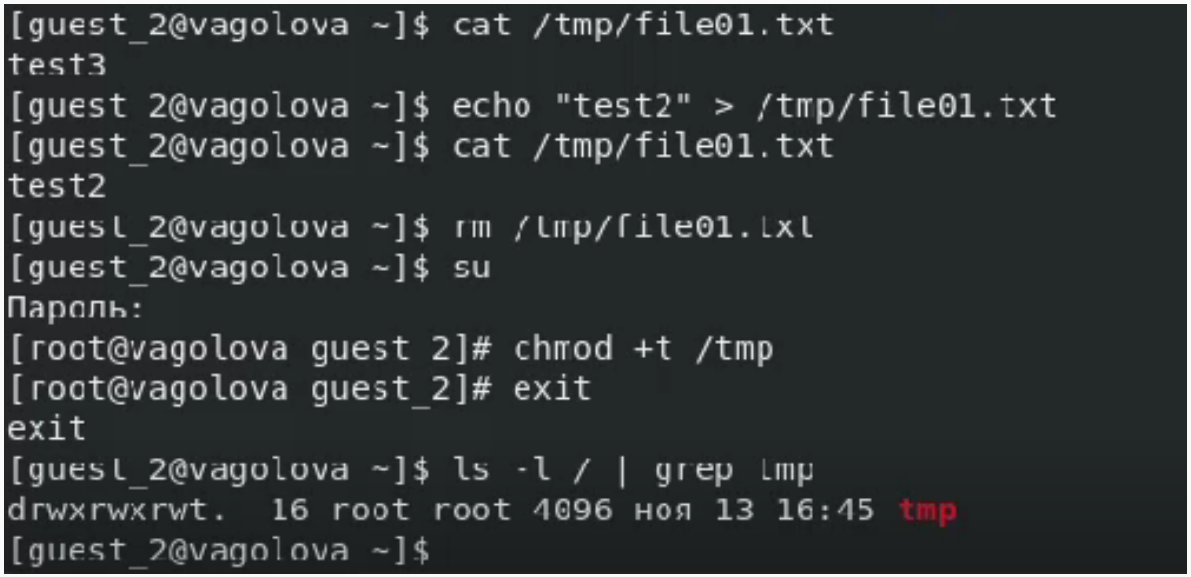


Figure 14: Атрибут t

Выводы

Выводы

Я 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.