Отчет по лабораторной работе №5

Основы информационной безопасности

Машков Илья Евгеньевич

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в кон- соли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

2 Выполнение лабораторной работы

Для лабораторной работы необходимо проверить, установлен ли компилятор gcc, комнда `gcc -v` позволяет это сделать. Также осуществляется отключение системы запретом с помощью `setenforce 0` (рис. 1).

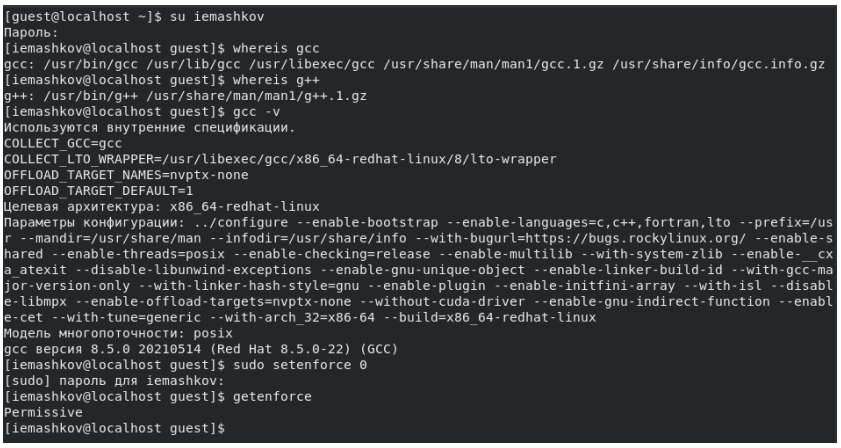

Рис. 2.1: Подготовка к лабораторной работе

Осуществляю вход от имени пользователя guest и создаю файл simpled.c и записываю в файл код (рис. 2)

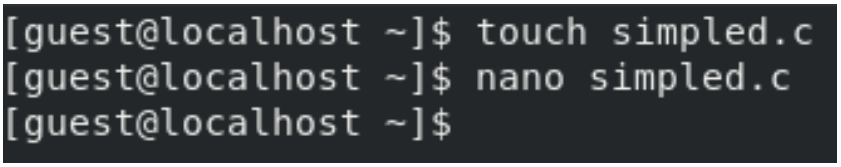

Рис. 2.2: Создание файла

C++ Листинг 1 #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; }

Содержимое файла выглядит следующти образом (рис. 3)

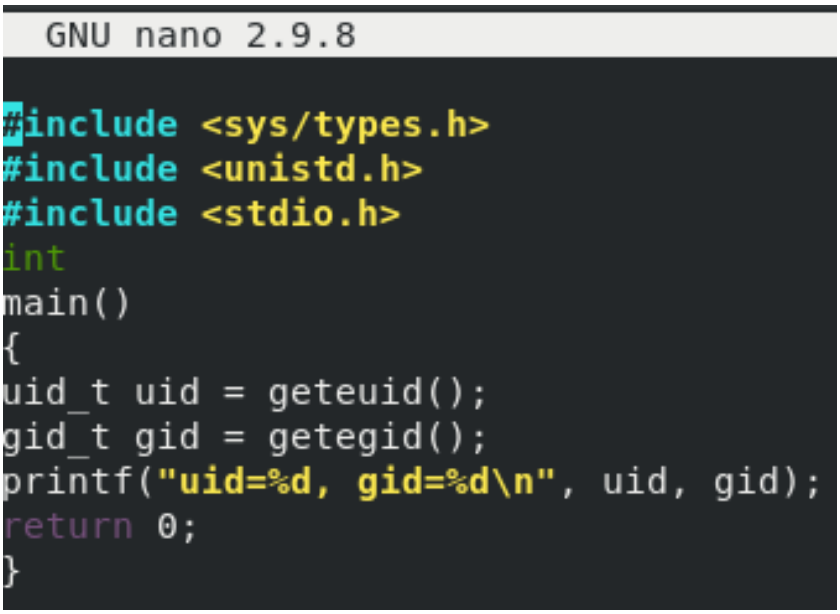

Рис. 2.3: Содержимое файла

Компилирую файл, проверяю, что он скомпилировался (рис. 4)

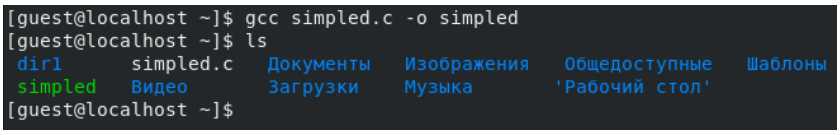

Рис. 2.4: Компиляция файла

Запускаю исполняемый файл. В выводе файла выписыны номера пользоватея и групп, от вывода при вводе if, они отличаются только тем, что информации меньше (рис. 5)

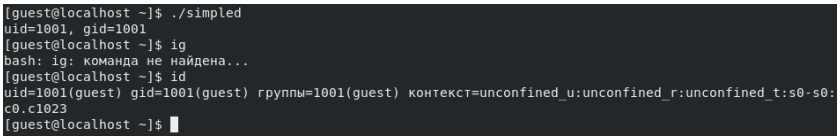

Рис. 2.5: Сравнение команд

Создание, запись в файл и компиляция файла simpled2.c. Запуск программы (рис. 6)

```
[guest@localhost ~]$ gcc simpled2.c -o simpled2
[guest@localhost ~]$ ./simpled2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@localhost ~]$
```

Рис. 2.6: Создание и компиляция файла

C++ Листинг 2 #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; }

(рис. 7)

```c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main()
{
uid_t real_uid = geteuid();
uid_t e_uid = geteuid();
gid_t real_gid = getegid();
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;
}
```

Рис. 2.7: Содержимое файла

С помощью chown изменяю владельца файла на суперпользователя, с помощью chmod изменяю права доступа (рис. 8)

```
[iemashkov@localhost guest]$ sudo chown root:guest /home/guest/simpled2
[sudo] пароль для iemashkov:
[iemashkov@localhost guest]$ sudo chmod u+s /home/guest/simpled2
[iemashkov@localhost guest]$ sudo ls -l /home/guest/simpled2
-rwsrwxr-x. 1 root guest 18208 апр 19 20:25 /home/guest/simpled2
[iemashkov@localhost guest]$
```

Рис. 2.8: Смена владельца файла и прав доступа к файлу

Сравнение вывода программы и команды id, наша команда снова вывела только ограниченное количество информации(рис. 9)

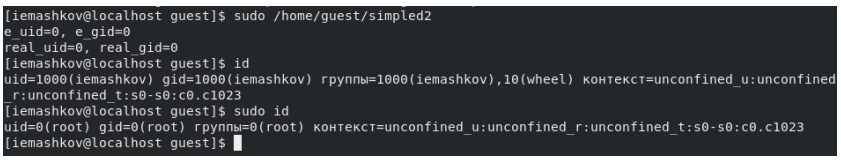

Рис. 2.9: Запуск файла

Создание и компиляция файла readfile.c (рис. 10)

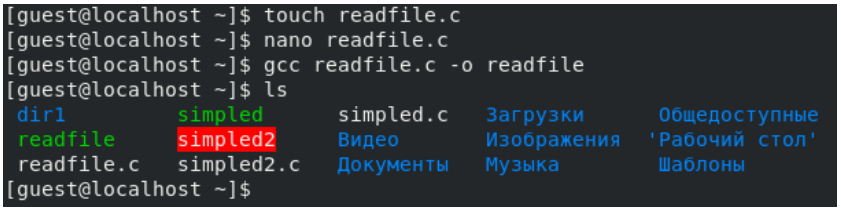

Рис. 2.10: Создание и компиляция файла

C++ Листинг 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; }

(рис. 11)

```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. 2.11: Содержимое файла

Снова от имени суперпользователи меняю владельца файла readfile. Далее меняю права доступа так, чтобы пользователь guest не смог прочесть содержимое файла (рис. 12)

```
[iemashkov@localhost guest]$ sudo chown root:guest /home/guest/readfile.c
[sudo] пароль для iemashkov:
[iemashkov@localhost guest]$ sudo chmod u+s /home/guest/readfile.c
[iemashkov@localhost guest]$ sudo chmod 700 /home/guest/readfile.c
[iemashkov@localhost guest]$ sudo chmod -r /home/guest/readfile.c
[iemashkov@localhost guest]$ sudo chmod u+s /home/guest/readfile.c
[iemashkov@localhost guest]$
```

Рис. 2.12: Смена владельца файла и прав доступа к файлу

Проверка прочесть файл от имени пользователя guest.Прочесть файл не удается (рис. 13)

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

Рис. 2.13: Попытка прочесть содержимое файла

Попытка прочесть тот же файл с помощью программы readfile, в ответ получаем “отказано в доступе” (рис. 14)

Рис. 2.14: Попытка прочесть содержимое файла программой

Попытка прочесть файл `\etc\shadow` с помощью программы, все еще получаем отказ в доступе (рис. 15)

Рис. 2.15: Попытка прочесть содержимое файла программой

Пробуем прочесть эти же файлы от имени суперпользователя и чтение файлов проходит успешно (рис. 16)

Рис. 2.16: Чтение файла от имени суперпользователя

Проверяем папку tmp на наличие атрибута Sticky, т.к. в выводе есть буква t, то атрибут установлен (рис. 17)

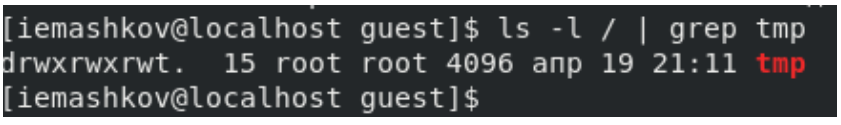

Рис. 2.17: Проверка атрибутов директории tmp

От имени пользователя guest создаю файл с текстом, добавляю права на чтение и запись для других пользователей (рис. 18)

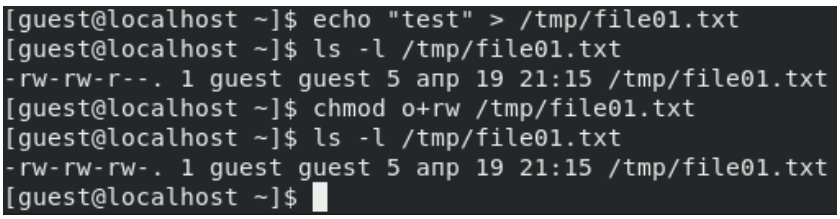

Рис. 2.18: Создание файла, изменение прав доступа

Вхожу в систему от имени пользователя guest2, от его имени могу и прочитать, и дозаписать файл file01.txt (рис. 19)

Рис. 2.19: Попытка чтения файла

Также могу добавить в файл file01.txt новую информацию от имени пользователя guest2 (рис. 20)

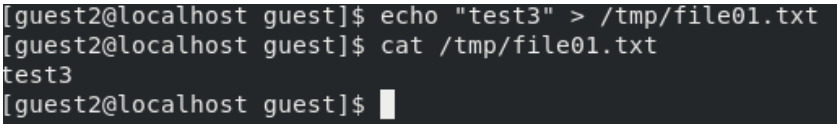

Рис. 2.20: Попытка записи в файл

Далее пробуем удалить файл, и получаем отказ (рис. 21)

Рис. 2.21: Попытка удалить файл

От имени суперпользователя снимаем с директории атрибут Sticky (рис. 22)

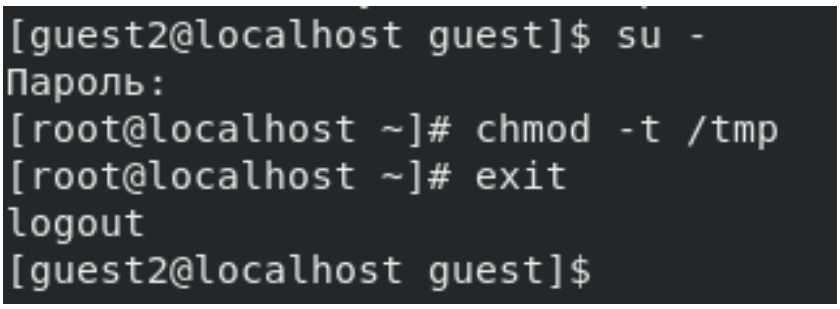

Рис. 2.22: Смена атрибутов файла

Проверяем, что атрибут действительно снят (рис. 23)

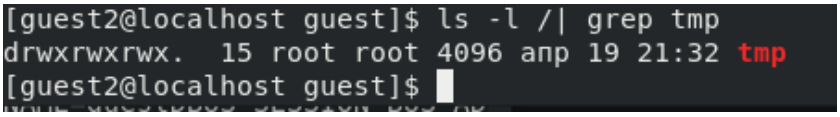

Рис. 2.23: Проверка атрибутов директории

Далее был выполнен повтор предыдущих действий. По результатам без Sticky-бита удаление файла, запись и дозапись в файл осталась возможной (рис. 24)

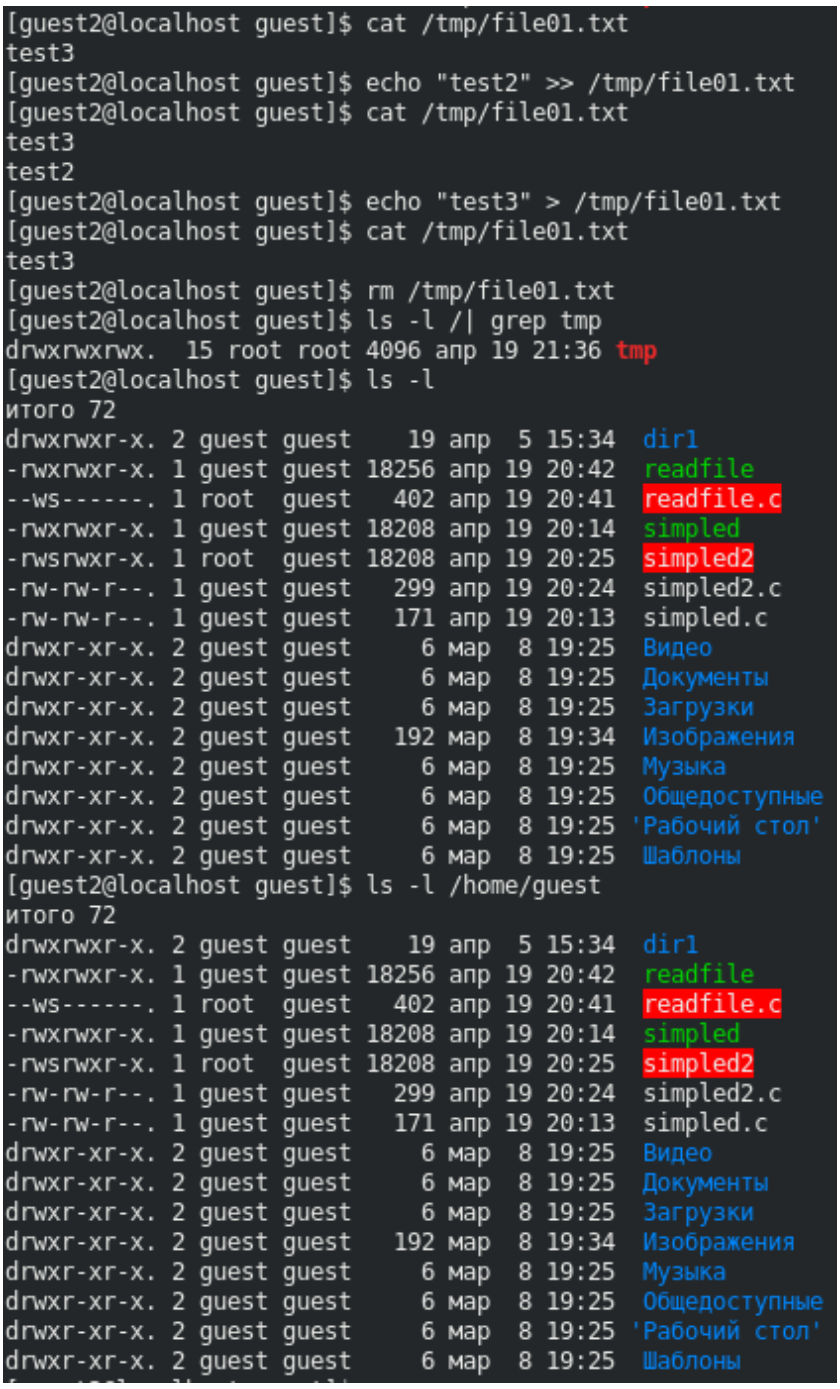

Рис. 2.24: Повтор предыдущих действий

Возвращение директории tmp атрибута t от имени суперпользователя (рис. 25)

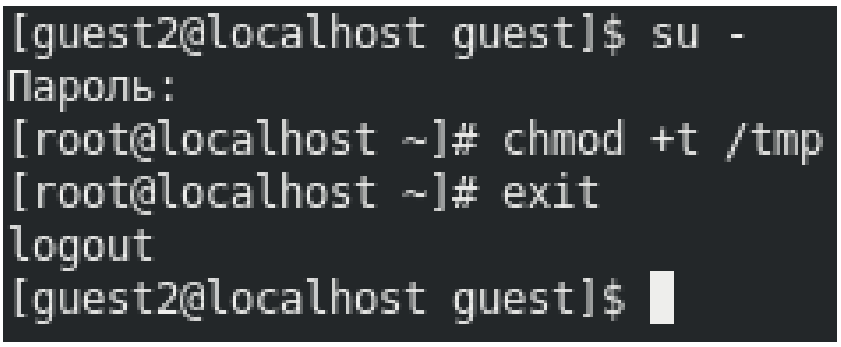

Рис. 2.25: Изменение атрибутов

3 Выводы

Изучил механизм изменения идентификаторов, применила SetUID- и Sticky-биты. 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

Основы информационной безопасности