Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Основы информационной безопасности

Симонова Полина Игоревна

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 Теоретическое введение

1. Дополнительные атрибуты файлов Linux

В Linux существует три основных вида прав — право на чтение (read), запись (write) и выполнение (execute), а также три категории пользователей, к которым они могут применяться — владелец файла (user), группа владельца (group) и все остальные (others). Но, кроме прав чтения, выполнения и записи, есть еще три дополнительных атрибута.

Sticky bit

Используется в основном для каталогов, чтобы защитить в них файлы. В такой каталог может писать любой пользователь. Но, из такой директории пользователь может удалить только те файлы, владельцем которых он является. Примером может служить директория /tmp, в которой запись открыта для всех пользователей, но нежелательно удаление чужих файлов.

SUID (Set User ID)

Атрибут исполняемого файла, позволяющий запустить его с правами владельца. В Linux приложение запускается с правами пользователя, запустившего указанное приложение. Это обеспечивает дополнительную безопасность т.к. процесс с правами пользователя не сможет получить доступ к важным системным файлам, которые принадлежат пользователю root.

SGID (Set Group ID)

Аналогичен suid, но относиться к группе. Если установить sgid для каталога, то все файлы созданные в нем, при запуске будут принимать идентификатор группы

каталога, а не группы владельца, который создал файл в этом каталоге.

Обозначение атрибутов sticky, suid, sgid

Специальные права используются довольно редко, поэтому при выводе программы ls -l символ, обозначающий указанные атрибуты, закрывает символ стандартных прав доступа.

Пример: `rwsrwsrwt`

где первая s — это suid, вторая s — это sgid, а последняя t — это sticky bit

В приведенном примере не понятно, rwt — это rw- или rwx? Определить это просто. Если t маленькое, значит x установлен. Если T большое, значит x не установлен. То же самое правило распространяется и на s.

В числовом эквиваленте данные атрибуты определяются первым символом при четырехзначном обозначении (который часто опускается при назначении прав), например в правах 1777 — символ 1 обозначает sticky bit.

2. Компилятор GCC

GCC - это свободно доступный оптимизирующий компилятор для языков C, C++. Собственно программа gcc это некоторая надстройка над группой компиляторов, которая способна анализировать имена файлов, передаваемые ей в качестве аргументов, и определять, какие действия необходимо выполнить. Файлы с расширением .cc или .C рассматриваются, как файлы на языке C++, файлы с расширением .c как программы на языке C, а файлы с расширением .o считаются объектными [**gcc**].

3 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).

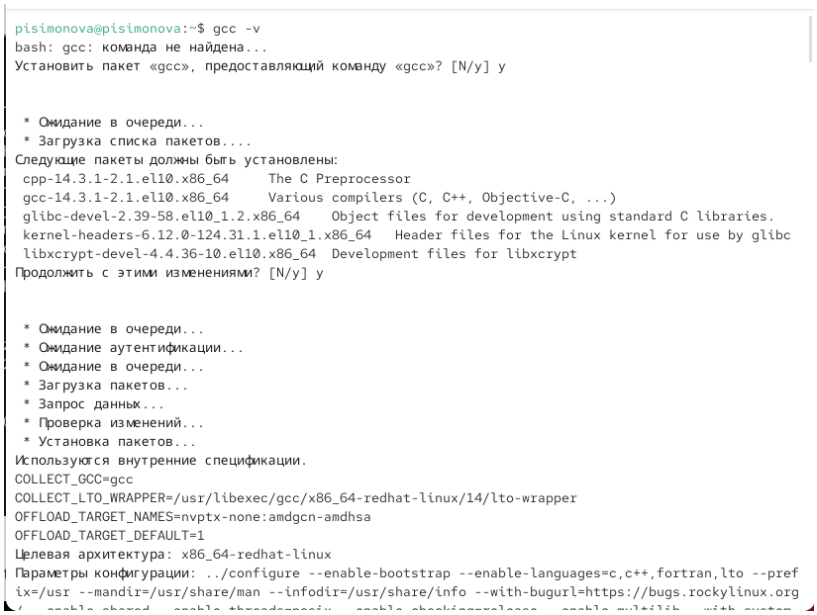

Рисунок 3.1: Подготовка к лабораторной работе

Осуществляется вход от имени пользователя guest (рис. 2).

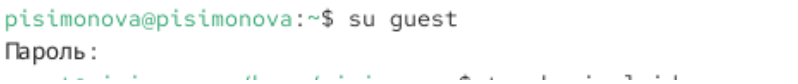

Рисунок 3.2: Вход от имени пользователя guest

Создание файла simpleid.c и запись в файл кода (рис. 3)

```
guest@pisimonova:~$ touch simpleid.c
guest@pisimonova:~$ nano simpleid.c
guest@pisimonova:~$ gcc simpleid.c -o simpleid
```

Рисунок 3.3: Создание файла

```
#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;
}
```

Листинг 1

Содержимое файла выглядит следующти образом (рис. 4)

Рисунок 3.4: Содержимое файла

Компилирую файл, проверяю, что он скомпилировался (рис. 5)

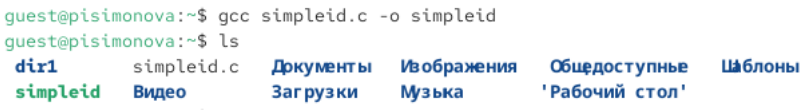

Рисунок 3.5: Компиляция файла

Запускаю исполняемый файл. В выводе файла выписаны номера пользоватея и групп, от вывода при вводе if, они отличаются только тем, что информации меньше (рис. 6)

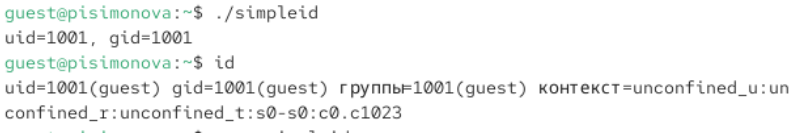

Рисунок 3.6: Сравнение команд

Создание, запись в файл и компиляция файла simpleid2.c. Запуск программы (рис. 7)

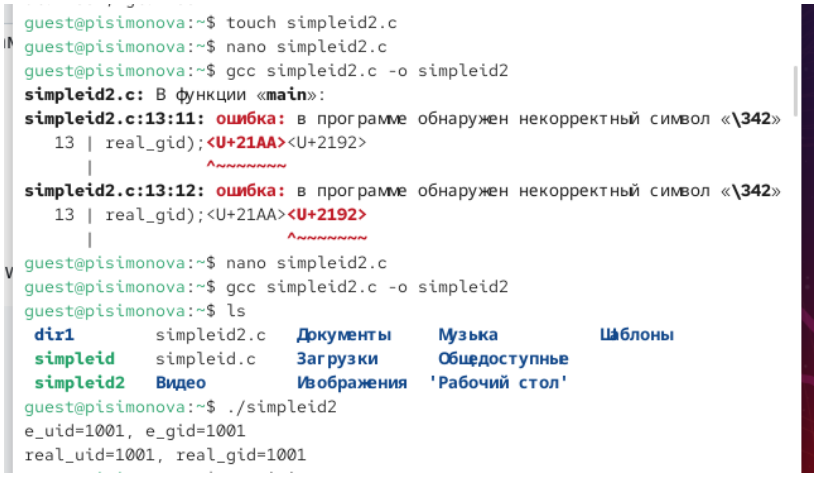

Рисунок 3.7: Создание и компиляция файла

```
#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;
}
```

Листинг 2

(рис. 8)

Рисунок 3.8: Содержимое файла

С помощью chown изменяю владельца файла на суперпользователя, с помощью chmod изменяю права доступа (рис. 9)

```
pisimonova@pisimonova:~$ sudo chown root:guest /home/guest/simpleid2
pisimonova@pisimonova:~$ sudo chmod u+s /home/guest/simpleid2


pisimonova@pisimonova:~$ sudo ls -l simpleid2
ls: невозможно получить доступ к 'simpleid2': Нет такого файла или каталога
pisimonova@pisimonova:~$ sudo ls -l /home/guest/simpleid2
-rwsr-xr-x. 1 root guest 16920 фев 14 14:38 /home/guest/simpleid2
```

Рисунок 3.9: Смена владельца файла и прав доступа к файлу

Сравнение вывода программы и команды id, наша команда снова вывела только ограниченное количество информации(рис. 10)

```
pisimonova@pisimonova:~$ sudo /home/guest/simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
pisimonova@pisimonova:~$ sudo id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:u
nconfined_t:s0-s0:c0.c1023
```

Рисунок 3.10: Запуск файла

Создание и компиляция файла readfile.c (рис. 11)

```
guest@pisimonova:~$ touch readfile.c
guest@pisimonova:~$ nano readfile.c
guest@pisimonova:~$ gcc readfile.c -o readfile
guest@pisimonova:~$ ls
 dir1        simpleid     simpleid.c   Загрузки      Общедоступные
 readfile    simpleid2    Видео        Изображения   'Рабочий стол'
 readfile.c  simpleid2.c  Документы    Музыка        Шаблоны
```

Рисунок 3.11: Создание и компиляция файла

```
#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;
}
```

Листинг 3

Рисунок 3.12: Содержимое файла

Снова от имени суперпользователи меняю владельца файла readfile. Далее меняю права доступа так, чтобы пользователь guest не смог прочесть содержимое файла (рис. 13)

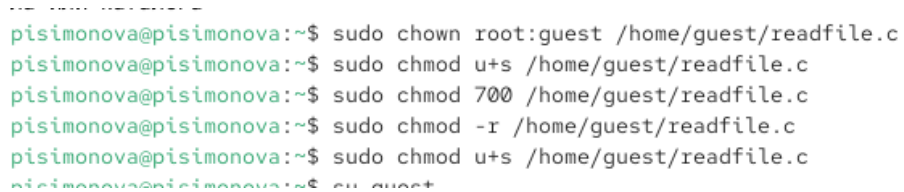

Рисунок 3.13: Смена владельца файла и прав доступа к файлу

Проверка прочесть файл от имени пользователя guest.Прочесть файл не удается (рис. 14)

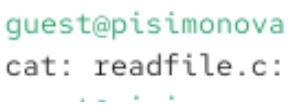

Рисунок 3.14: Попытка прочесть содержимое файла

Попытка прочесть тот же файл с помощью программы readfile, в ответ получаем «отказано в доступе» (рис. 15)

Рисунок 3.15: Попытка прочесть содержимое файла программой

Попытка прочесть файл `\etc\shadow` с помощью программы, все еще получаем отказ в доступе (рис. 16)

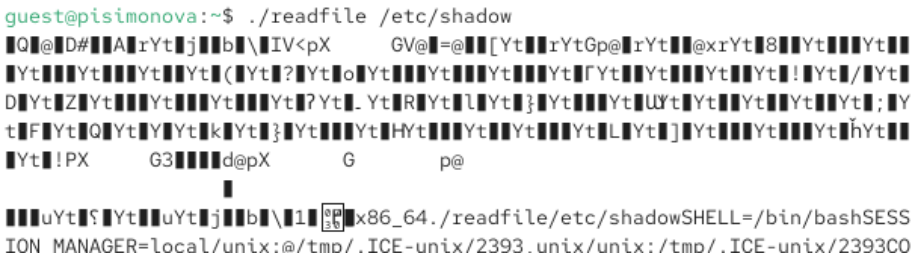

Рисунок 3.16: Попытка прочесть содержимое файла программой

Пробуем прочесть эти же файлы от имени суперпользователя и чтение файлов проходит успешно (рис. 17)

```
pisimonova@pisimonova:~$ sudo /home/guest/readfile /etc/shadow
root:$y$j9T$szYenPDDQgj0aE34FNj70aUF$gv8u9dZtGzaXbu1lf9KN3eK75PsAKe2BMP7xjB
BGibD::0:99999:7:::
bin:*:20186:0:99999:7:::
daemon:*:20186:0:99999:7:::
adm:*:20186:0:99999:7:::
lp:*:20186:0:99999:7:::
sync:*:20186:0:99999:7:::
shutdown:*:20186:0:99999:7:::
```

Рисунок 3.17: Чтение файла от имени суперпользователя

Проверяем папку tmp на наличие атрибута Sticky, т.к. в выводе есть буква t, то атрибут установлен (рис. 18)

```
pisimonova@pisimonova:~$ sudo ls -l / | grep tmp
drwxrwxrwt.  18 root root 4096 фев 14 14:54 tmp
```

Рисунок 3.18: Проверка атрибутов директории tmp

От имени пользователя guest создаю файл с текстом, добавляю права на чтение и запись для других пользователей (рис. 19)

```
guest@pisimonova:~$ echo "test" > /tmp/file01.txt
guest@pisimonova:~$ ls
 dir1        simpleid     simpleid.c   Загрузки      Общедоступные
 readfile    simpleid2    Видео        Изображения  'Рабочий стол'
 readfile.c  simpleid2.c  Документы    Музыка        Шаблоны
guest@pisimonova:~$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 фев 14 14:58 /tmp/file01.txt
guest@pisimonova:~$ chmod o+rw /tmp/file01.txt
guest@pisimonova:~$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 фев 14 14:58 /tmp/file01.txt
```

Рисунок 3.19: Создание файла, изменение прав доступа

Вхожу в систему от имени пользователя guest2, от его имени могу прочитать файл file01.txt, но перезаписать информацию в нем не могу (рис. 20)

```
guest@pisimonova:~$ su guest2
Пароль:
guest2@pisimonova:/home/guest$ cd
guest2@pisimonova:~$ cat /tmp/file01.txt
test
guest2@pisimonova:~$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
guest2@pisimonova:~$ cat /tmp/file01.txt
test
```

Рисунок 3.20: Попытка чтения файла

Также невозможно добавить в файл file01.txt новую информацию от имени пользователя guest2 (рис. 21)

```
guest2@pisimonova:~$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
guest2@pisimonova:~$ cat /tmp/file01.txt
test
```

Рисунок 3.21: Попытка записи в файл

Далее пробуем удалить файл, снова получаем отказ (рис. 22)

```
guest2@pisimonova:~$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
```

Рисунок 3.22: Попытка удалить файл

От имени суперпользователя снимаем с директории атрибут Sticky (рис. 23)

```
guest2@pisimonova:~$ su -
Пароль:
root@pisimonova:~# chmod -t /tmp
root@pisimonova:~# exit
выход
guest2@pisimonova:~$ ls -l / | grep tmp
```

Рисунок 3.23: Смена атрибутов файла

Проверяем, что атрибут действительно снят (рис. 24)

```
guest2@pisimonova:~$ ls -l / | grep tmp
drwxrwxrwx.  18 root root 4096 фев 14 15:04 tmp
```

Рисунок 3.24: Проверка атрибутов директории

Далее был выполнен повтор предыдущих действий. По результатам без Sticky-бита запись в файл и дозапись в файл осталась невозможной, зато удаление файла прошло успешно (рис. 25)

```
guest2@pisimonova:~$ cat /tmp/file01.txt
test
guest2@pisimonova:~$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
guest2@pisimonova:~$ cat /tmp/file01.txt
test
guest2@pisimonova:~$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
guest2@pisimonova:~$ cat /tmp/file01.txt
test
guest2@pisimonova:~$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
guest2@pisimonova:~$ ls -l / | grep tmp
drwxrwxrwx.  18 root root 4096 фев 14 15:05 tmp
guest2@pisimonova:~$ ls -l
итого 0
guest2@pisimonova:~$ ls -l /home/guest
итого 72
drwx------. 2 guest guest    19 фев 14 13:59  dir1
-rwxr-xr-x. 1 guest guest 16864 фев 14 14:48  readfile
--ws------. 1 root  guest   402 фев 14 14:47  readfile.c
-rwxr-xr-x. 1 guest guest 16816 фев 14 14:35  simpleid
-rwsr-xr-x. 1 root  guest 16920 фев 14 14:38  simpleid2
-rw-r--r--. 1 guest guest   303 фев 14 14:38  simpleid2.c
-rw-r--r--. 1 guest guest   175 фев 14 14:35  simpleid.c
drwxr-xr-x. 2 guest guest     6 фев 13 21:34  Видео
drwxr-xr-x. 2 guest guest     6 фев 13 21:34  Документы
drwxr-xr-x. 2 guest guest     6 фев 13 21:34  Загрузки
drwxr-xr-x. 2 guest guest     6 фев 13 21:34  Изображения
drwxr-xr-x. 2 guest guest     6 фев 13 21:34  Музыка
drwxr-xr-x. 2 guest guest     6 фев 13 21:34  Общедоступные
drwxr-xr-x. 2 guest guest     6 фев 13 21:34 'Рабочий стол'
drwxr-xr-x. 2 guest guest     6 фев 13 21:34  Шаблоны
```

Рисунок 3.25: Повтор предыдущих действий

Возвращение директории tmp атрибута t от имени суперпользователя (рис. 26)

```
root@pisimonova:~# chmod +t /tmp
root@pisimonova:~# exut
bash: exut: команда не найдена...
exit
root@pisimonova:~# exit
выход
```

Рисунок 3.26: Изменение атрибутов

4 Выводы

Изучила 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.

5 Список литературы

[0] Методические материалы курса

[1] Права доступа: https://codechick.io/tutorials/unix-linux/unix-linux-permissions