Лабораторная работа №5

Информационная безопасность

Доре Стевенсон Эдгар

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.

Задание

Выполнение лабораторной работы

Установил gcc с помощью команды `yum install gcc`. (рис. @fig:001)

```
mint@mint-VirtualBox:~/Рабочий стол$ sudo apt install gcc
[sudo] пароль для mint:
Чтение списков пакетов… Готово
Построение дерева зависимостей
Чтение информации о состоянии… Готово
Уже установлен пакет gcc самой новой версии (4:9.3.0-1ubuntu2).
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакето
в, и 243 пакетов не обновлено.
mint@mint-VirtualBox:~/Рабочий стол$ setenforce 0
```

Установка gcc

Отменил на текущую сессию SELinux командой `setenforce 0`. Вошёл в систему от имени пользователя guest, создал программу `simpleid.c`. (рис. @fig:002)

```
 GNU nano 4.8                          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;
}
```

Код программы `simpleid.c`

Скомпилировал программу и убедился, что файл программы создан: `gcc simpleid.c -o simpleid`. Выполнил программу simpleid: `./simpleid`. Выполнил программу `id` и сравнил полученный результат с данными предыдущего пункта задания. Полученные значения id совпадают. (рис. @fig:003)

```
$ nano simpleid.c
$ gcc simpleid.c -o simpleid
$ ./simpleid
uid=1001, gid=1001
$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest)
$
```

Сравнение результатов программы и команды

Усложнил программу, добавив вывод действительных идентификаторов, получившуюися программу назвал `simpleid2.c`. (рис. @fig:004)

```
  GNU nano 4.8                          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_uid=%d\n", real_uid, real_gid);
        return 0;
}
```

Код программы `simpleid2.c`

Скомпилировал и запустил simpleid2.c `gcc simpleid2.c -o simpleid2`, а затем `./simpleid2`. (рис. @fig:005)

```
$ nano simpleid2.c
$ gcc simpleid2.c -o simpleid2
$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_uid=1001
```

Компиляция и запуск `simpleid2.c`

От имени суперпользователя выполнил команды: `chown root:guest /home/guest/simpleid2`, а затем `chmod u+s /home/guest/simpleid2`. Первая команда изменяет права на файл с guest на root. А затем устанавливает атрибут SetUID, который запускает программу не с правами пользователя, а с правами владельца файла. Затем выполнил проверку изменений с помощью команды `ls -l simpleid2`. (рис. @fig:006)

```
Настраивается пакет libc6-dbg:amd64 (2.31-0ubuntu9.2) …
Настраивается пакет libcrypt-dev:amd64 (1:4.4.10-10ubuntu4) …
Настраивается пакет libc-dev-bin (2.31-0ubuntu9.2) …
Настраивается пакет libc6-dev:amd64 (2.31-0ubuntu9.2) …
Обрабатываются триггеры для man-db (2.9.1-1) …
Обрабатываются триггеры для libc-bin (2.31-0ubuntu9.1) …
mint@mint-VirtualBox:~/Рабочий стол$ sudo chown root:guest /home/guest/simpleid2
mint@mint-VirtualBox:~/Рабочий стол$ sudo chmod u+s /home/guest/simpleid2
mint@mint-VirtualBox:~/Рабочий стол$
```

Добавление SetUID

Запустил simpleid2 и id: `./simpleid2`, `id`. При данном запуску выводы совпадают. (рис. @fig:007)

```
$ ls -l simpleid2
-rwsrwxr-x 1 root guest 16880 ноя 13 18:43 simpleid2
$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_uid=1001
$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest)
$
```

Сверка результата программы и кода

Проделал то же самое с атрибутом SetGID (установление прав для владеющей группы). Запустил файл. Теперь выводы для группы различны.

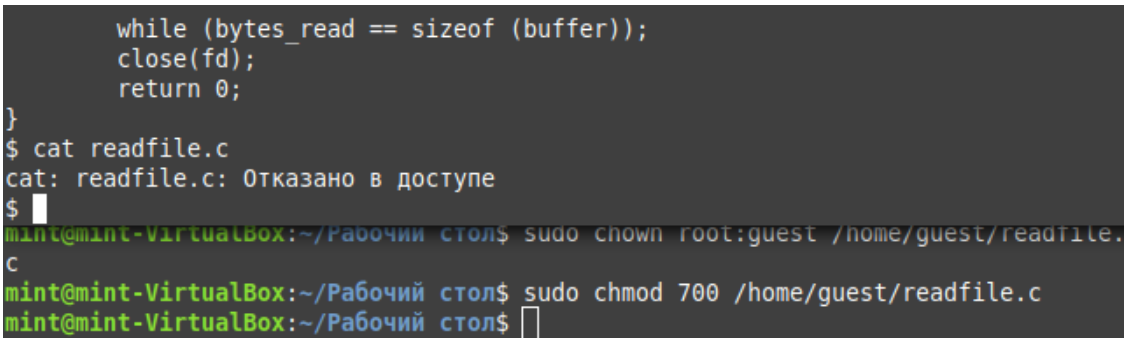
Создал программу `readfile.c`. (рис. @fig:008)

```
 GNU nano 4.8                    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;
}
```

Код программы readfile.c

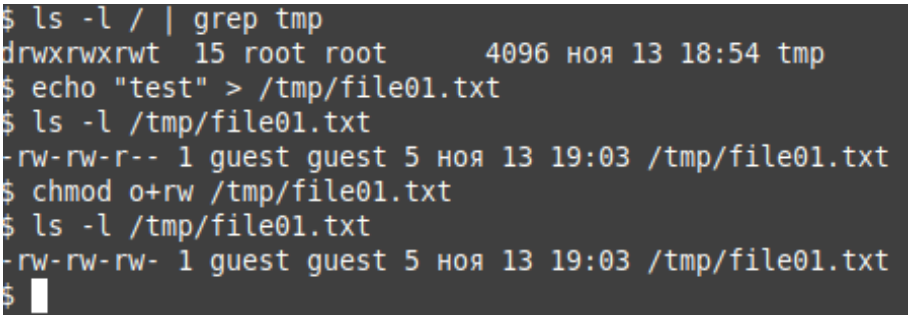
Откомпилировал программу: `gcc readfile.c -o readfile`. Сменил владельца у файла readfile.c и изменил права так, чтобы только суперпользователь(root) мог

прочитать его, а guest не мог. Проверил, что пользователь guest не может прочитать файл readfile.c (рис. @fig:009)



Проверка чтения файла

Сменил у программы readfile владельца и установил SetU'D-бит. Программа readfile может прочитать файл readfile.c. Программа readfile может прочитать файл /etc/shadow. Исследование Sticky-бита. Узнал, установлен ли атрибут Sticky на директории /tmp, для чего выполнил команду `ls -l / | grep tmp` (рис. @fig:010)



Проверка атрибутов

От имени пользователя guest создал файл file01.txt в директории /tmp со словом test `echo "test" > /tmp/file01.txt`. Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей «все остальные»: `ls -l /tmp/file01.txt`, `chmod o+rw /tmp/file01.txt`, `ls -l /tmp/file01.txt`. От пользователя guest2 (не являющегося владельцем) попробовал прочитать файл /tmp/file01.txt: `cat /tmp/file01.txt`, записать в файл `/tmp/file01.txt` текст test3, стерев при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt`. Проверил содержимое файла командой `cat /tmp/file01.txt`, попробовал дозаписать в файл `/tmp/file01.txt` слово test2 командой `echo "test2" >> /tmp/file01.txt`, удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt` Файл удалить не удалось. (рис. @fig:011)

```
mint@mint-VirtualBox:~/Рабочий стол$ su - guest2
Пароль:
$ cat /tmp/file01.txt
test
$ echo "test2" > /tmp/file01.txt
-sh: 2: cannot create /tmp/file01.txt: Permission denied
$ echo "test3" > /tmp/file01.tx
$ echo "test3" > /tmp/file01.txt
-sh: 4: cannot create /tmp/file01.txt: Permission denied
$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
$ su -
```

Проверка от guest2

Повысил свои права до суперпользователя следующей командой `su -` и выполнил после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: `chmod -t /tmp`. Затем попробовал выполнить все вышеперечисленные операции. Все удалось. (рис. @fig:012)

```
$ ls -l / | grep tmp
drwxrwxrwx  15 root root      4096 ноя 13 19:07 tmp
$ echo "test2" > /tmp/file01.txt
$ cat /tmp/file01.txt
test2
$ rm /tmp/file01.txt
$
```

Проверка после снятия Sticky атрибута

Повысил свои права до суперпользователя и вернул атрибут `t` на директорию /tmp: `su -`, `chmod +t /tmp`, `exit`.

Выводы

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