

Отчёт по лабораторной работе №5

Информационная безопасность

**Дискреционное разграничение прав в Linux. Исследование
влияния дополнительных атрибутов**

Хусаинова Динара Айратовна,
НПИбд-02-21, 1032212283

Содержание

Цель работы	4
Теоретическое введение	5
Выполнение лабораторной работы	7
Исследование Sticky-бита	10
Выводы	13
Список литературы. Библиография	14

Список иллюстраций

1	Создание файла	7
2	Компиляция	7
3	Сравнения вывода команды и кода из файла	7
4	Усложненная программа, код	8
5	От имени суперпользователя команды	8
6	simpleid2 и id	9
7	SetGID-бит	9
8	Изменение прав	10
9	Проверка возможности чтение файла	10
10	Выполнение действий	12

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.

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. [[@u](#)]

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 относится 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: `rws rws rwt`

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`. Остальные атрибуты имеют следующие числовое соответствие:

1 — установлен `sticky bit`

2 — установлен `sgid`

4 — установлен `suid`

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]`.

Выполнение лабораторной работы

1. Войдите в систему от имени пользователя guest.
2. Создайте программу simpleid.c:

```
[guest1@dakhusainova ~]$ touch simplified.c  
[guest1@dakhusainova ~]$ mcedit simplified.c
```

Рис. 1: Создание файла

3. Скомпилируйте программу и убедитесь, что файл программы создан: gcc simpleid.c -o simpleid

```
[guest1@dakhusainova ~]$ gcc simplified.c -o simplified  
[guest1@dakhusainova ~]$ ls  
Desktop Documents Music Public simplified.c Videos  
dir1 Downloads Pictures simplified Templates  
[guest1@dakhusainova ~]$
```

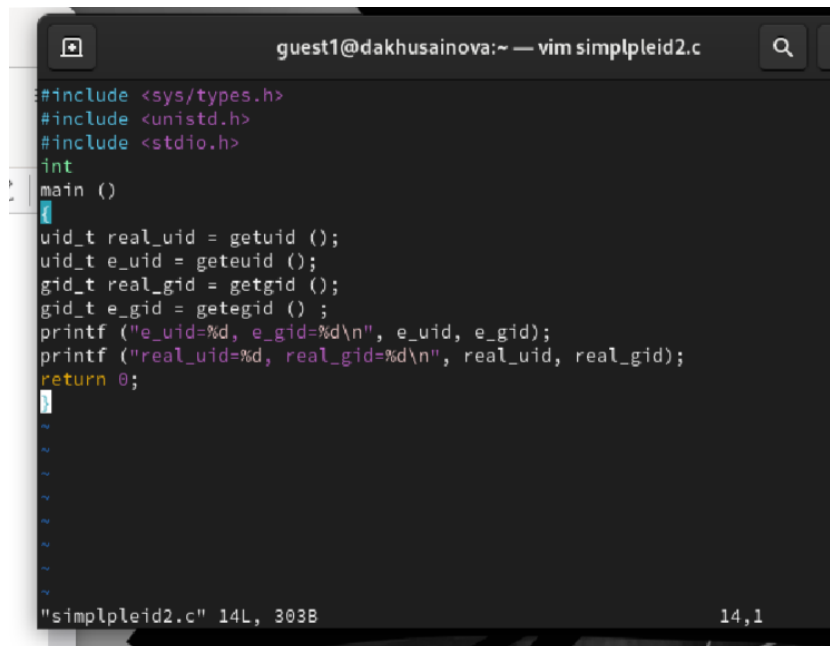
Рис. 2: Компиляция

4. Выполните программу simpleid: ./simpleid
5. Выполните системную программу id: id и сравните полученный вами результат с данными предыдущего пункта задания

```
[guest1@dakhusainova ~]$ ./simplified  
uid=1003, gid=1003  
[guest1@dakhusainova ~]$ id  
uid=1003(guest1) gid=1003(guest1) groups=1003(guest1) context=unconfined_u:unc
```

Рис. 3: Сравнения вывода команды и кода из файла

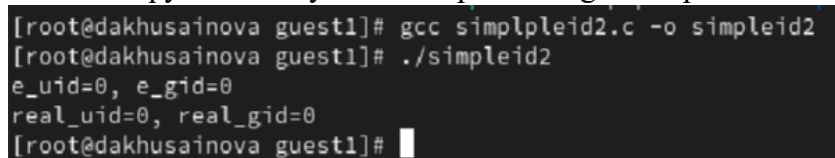
6. Усложните программу, добавив вывод действительных идентификаторов



```
#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;
}
```

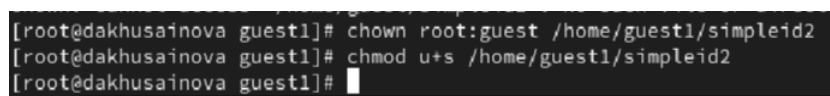
Рис. 4: Усложненная программа, код

7. Скомпилируйте и запустите simpleid2.c: `gcc simpleid2.c -o simpleid2 ./simpleid2`



```
[root@dakhusainova guest1]# gcc simpleid2.c -o simpleid2
[root@dakhusainova guest1]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@dakhusainova guest1]#
```

8. От имени суперпользователя выполните команды:



```
[root@dakhusainova guest1]# chown root:guest /home/guest1/simpleid2
[root@dakhusainova guest1]# chmod u+s /home/guest1/simpleid2
[root@dakhusainova guest1]#
```

Рис. 5: От имени суперпользователя команды

9. Используйте `sudo` или повысьте временно свои права с помощью `su`. Поясните, что делают эти команды.

10. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2: `ls -l simpleid2`
11. Запустите simpleid2 и id: `./simpleid2 id`

```
[root@dakhusainova guest1]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 17720 Oct  1 18:39 simpleid2
[root@dakhusainova guest1]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@dakhusainova guest1]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@dakhusainova guest1]#
```

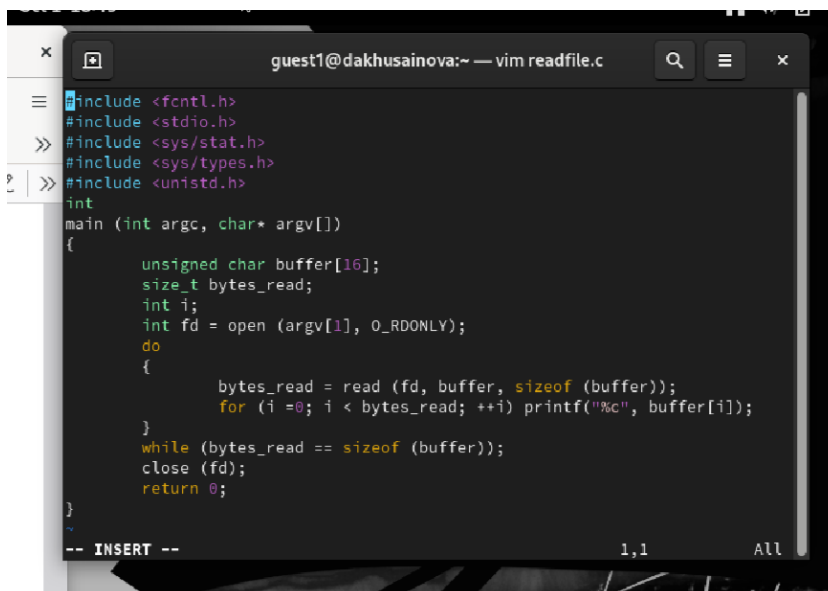
Рис. 6: simpleid2 и id

12. Прodelайте тоже самое относительно SetGID-бита.

```
[root@dakhusainova guest1]# sudo id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@dakhusainova guest1]# su guest1
[guest1@dakhusainova ~]$ id
uid=1003(guest1) gid=1003(guest1) groups=1003(guest1) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest1@dakhusainova ~]$
```

Рис. 7: SetGID-бит

13. Создайте программу readfile.c



```
guest1@dakhusainova:~ — vim 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;
}
-- INSERT -- 1,1 All
```

14. Откомпилируйте её. `gcc readfile.c -o readfile`
15. Смените владельца у файла `readfile.c` (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (`root`) мог прочитать его, а `guest` не мог.

```
[root@dakhusainova guest1]# sudo chmod 700 /home/guest1/dir1
[root@dakhusainova guest1]# sudo chmod 700 /home/guest1/readfile.c
[root@dakhusainova guest1]# sudo chmod -r /home/guest1/readfile.c
[root@dakhusainova guest1]# sudo chmod u+s /home/guest1/readfile.c
[root@dakhusainova guest1]#
```

Рис. 8: Изменение прав

16. Проверьте, что пользователь `guest` не может прочитать файл `readfile.c`.
17. Смените у программы `readfile` владельца и установите SetU'D-бит.
18. Проверьте, может ли программа `readfile` прочитать файл `readfile.c`?
19. Проверьте, может ли программа `readfile` прочитать файл `/etc/shadow`?

```
[root@dakhusainova guest1]# su guest1
[guest1@dakhusainova ~]$ cat readfile.c
cat: readfile.c: Permission denied
[guest1@dakhusainova ~]$
```

Рис. 9: Проверка возможности чтения файла

Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории `/tmp`, для чего выполните команду `ls -l / | grep tmp`
2. От имени пользователя `guest` создайте файл `file01.txt` в директории `/tmp` со словом `test`: `echo "test" > /tmp/file01.txt`
3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»: `ls -l /tmp/file01.txt chmod o+rw /tmp/file01.txt ls -l /tmp/file01.txt`

4. От пользователя `guest2` (не являющегося владельцем) попробуйте прочитать файл `/tmp/file01.txt`: `cat /tmp/file01.txt`
5. От пользователя `guest2` попробуйте дозаписать в файл `/tmp/file01.txt` слово `test2` командой `echo "test2" > /tmp/file01.txt` Удалось ли вам выполнить операцию?
6. Проверьте содержимое файла командой `cat /tmp/file01.txt`
7. От пользователя `guest2` попробуйте записать в файл `/tmp/file01.txt` слово `test3`, стерев при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt` Удалось ли вам выполнить операцию?
8. Проверьте содержимое файла командой `cat /tmp/file01.txt`
9. От пользователя `guest2` попробуйте удалить файл `/tmp/file01.txt` командой `rm /tmp/file01.txt` Удалось ли вам удалить файл?
10. Повысьте свои права до суперпользователя следующей командой `su -` и выполните после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`: `chmod -t /tmp`
11. Покиньте режим суперпользователя командой `exit`
12. От пользователя `guest2` проверьте, что атрибута `t` у директории `/tmp` нет: `ls -l / | grep tmp`
13. Повторите предыдущие шаги. Какие наблюдаются изменения?
14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт.
15. Повысьте свои права до суперпользователя и верните атрибут `t` на директорию `/tmp`: `su - chmod +t /tmp exit`

```
guest2@dakhusainova:/home/guest1
[guest1@dakhusainova ~]$ ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 Oct  1 18:52 tmp
[guest1@dakhusainova ~]$ echo "test" > /tmp/file01.txt
[guest1@dakhusainova ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest1 guest1 5 Oct  1 18:57 /tmp/file01.txt
[guest1@dakhusainova ~]$ chmod o+rw /tmp/file01.txt
[guest1@dakhusainova ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest1 guest1 5 Oct  1 18:57 /tmp/file01.txt
[guest1@dakhusainova ~]$ su guest2
Password:
[guest2@dakhusainova guest1]$ cat /tmp/file01.txt
test
[guest2@dakhusainova guest1]$ echo "test2" > /tmp/file01.txt
[guest2@dakhusainova guest1]$ cat /tmp/file01.txt
test2
[guest2@dakhusainova guest1]$ echo "test3" > /tmp/file01.txt
[guest2@dakhusainova guest1]$ cat /tmp/file01.txt
test3
[guest2@dakhusainova guest1]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': No such file or directory
[guest2@dakhusainova guest1]$ su -
Password:
[root@dakhusainova ~]# chmod -t /tmp
[root@dakhusainova ~]# exit
logout
[guest2@dakhusainova guest1]$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 Oct  1 19:00 tmp
[guest2@dakhusainova guest1]$ echo "test2" > /tmp/file01.txt
[guest2@dakhusainova guest1]$ cat /tmp/file01.txt
test2
[guest2@dakhusainova guest1]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': No such file or directory
[guest2@dakhusainova guest1]$ su -
Password:
[root@dakhusainova ~]# chmod +t /tmp
[root@dakhusainova ~]# exit
logout
[guest2@dakhusainova guest1]$
```

Рис. 10: Выполнение действий

Выводы

Изучила механизм изменения идентификаторов, применила SetUID- и Sticky-биты. Получила практические навыки работы в кон- соли с дополнительными атрибутами. Рассмотрела работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Список литературы. Библиография

[0] Методические материалы курса

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

[2] Расширенные атрибуты: <https://ru.manpages.org/xattr/7>

[3] Операции с расширенными атрибутами: <https://p-n-z-8-8.livejournal.com/64493.html>