

Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов.

Пономарева Лилия Михайловна

Содержание

Цель работы	4
Теоретическое введение	5
Выполнение лабораторной работы	7
Выводы	19
Список литературы	20

Список иллюстраций

1	Вход под пользователем guest	7
2	Код программы simpleid.c	7
3	Установка расширенного атрибута от имени обычного пользователя	8
4	Программа simpleid2.c	8
5	Вывод реальных и действительных идентификаторов	9
6	Смена владельца файла и прав доступа	9
7	Вывод программы simpleid2	9
8	SetGID-бит	10
9	Код программы readfile.c	11
10	Попытка изменения прав доступа файла	11
11	Смена владельца файла	12
12	Смена атрибутов	12
13	Проверка доступа к файлу	13
14	Смена владельца и добавление SetUID у readfile	13
15	Чтение файла readfile.c программой readfile	14
16	Чтение файла /etc/shadow программой из readfile	15
17	Атрибут Sticky на директории /tmp	15
18	Создание файла file01.txt в директории /tmp	15
19	Чтение и запись для категории пользователей «все остальные» . .	16
20	Чтение файла /tmp/file01.txt	16
21	Дозапись файла /tmp/file01.txt	16
22	Перезапись файла /tmp/file01.txt	17
23	Попытка удаления файла /tmp/file01.txt	17
24	Снятие Sticky-бит	17
25	Проверка атрибутов	17
26	Проверка атрибутов	18

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 [1].

Теоретическое введение

Биты SUID, SGID и Sticky.

Unix отслеживает не символьные имена владельцев и групп, а их идентификаторы (UID - для пользователей и GID для групп). Эти идентификаторы хранятся в файлах `/etc/passwd` и `/etc/group` соответственно. Символьные эквиваленты идентификаторов используются только для удобства, например, при использовании команды `ls`, идентификаторы заменяются соответствующими символьными обозначениями.

Что касается процессов, то с ними связано не два идентификатора, а 4-е: реальный и эффективный пользовательский (UID), а также реальный и эффективный групповой (GID). Реальные номера применяются для учета использования системных ресурсов, а эффективные для определения прав доступа к процессам. Как правило, реальные и эффективные идентификаторы совпадают. Владелец процесса может посылать ему сигналы, а также изменять приоритет.

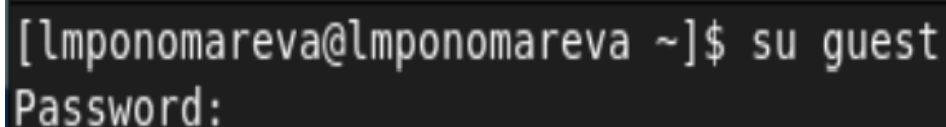
Процесс не может явно изменить ни одного из своих четырех идентификаторов, но есть ситуации когда происходит косвенная установка новых эффективных идентификаторов процесса. Дело в том, что существуют два специальных бита: SUID (Set User ID - бит смены идентификатора пользователя) и SGID (Set Group ID - бит смены идентификатора группы). Когда пользователь или процесс запускает исполняемый файл с установленным одним из этих битов, файлу временно назначаются права его (файла) владельца или группы (в зависимости от того, какой бит задан). Таким образом, пользователь может даже запускать файлы от имени суперпользователя.

Вобщем, одним словом установка битов SUID или SGID позволит пользователям запускать исполняемые файлы от имени владельца (или группы) запускаемого файла. Например, как говорилось выше, команду `chmod` по умолчанию может запускать только `root`. Если мы установим SUID на исполняемый файл `/bin/chmod`, то обычный пользователь сможет использовать эту команду без использования `sudo`, так, что она будет выполняться от имени пользователя `root`. В некоторых случаях очень удобное решение. Кстати по такому принципу работает команда `passwd`, с помощью которой пользователь может изменить свой пароль.

Еще одно важное усовершенствование касается использования `sticky`-бита в каталогах. Каталог с установленным `sticky`-битом означает, что удалить файл из этого каталога может только владелец файла или суперпользователь. Другие пользователи лишаются права удалять файлы. Установить `sticky`-бит в каталоге может только суперпользователь. `Sticky`-бит каталога, в отличие от `sticky`-бита файла, остается в каталоге до тех пор, пока владелец каталога или суперпользователь не удалит каталог явно или не применит к нему `chmod`. Заметьте, что владелец может удалить `sticky`-бит, но не может его установить.

Выполнение лабораторной работы

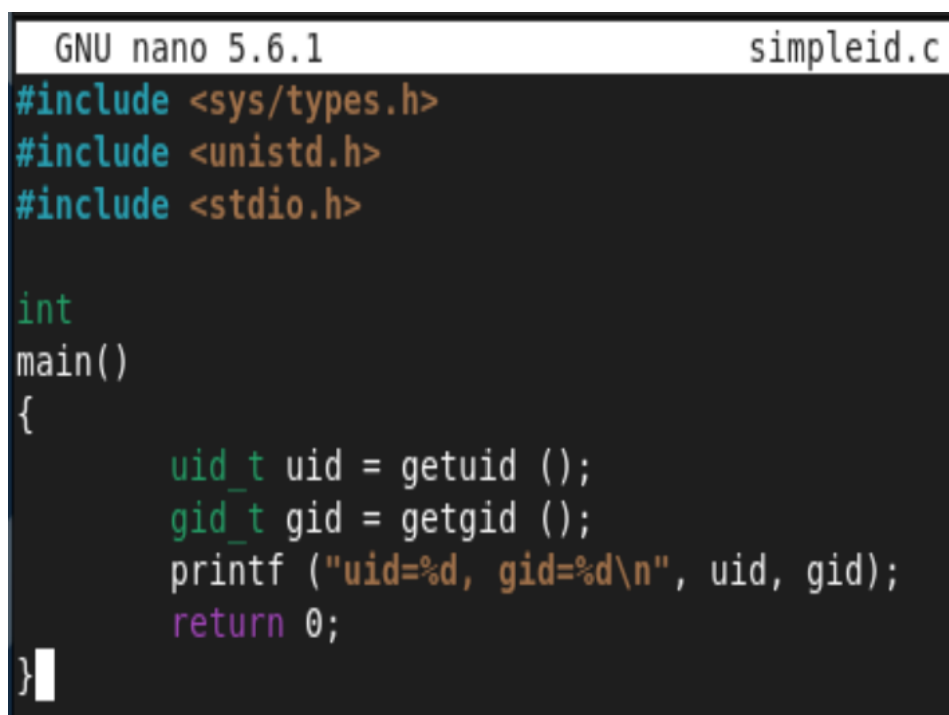
1. Вошла в систему от имени пользователя guest(рис. 1).



```
[lmponomareva@lmponomareva ~]$ su guest
Password:
```

Рис. 1: Вход под пользователем guest

2. Создала программу simpleid.c (рис. 2).



```
GNU nano 5.6.1 simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main()
{
    uid_t uid = getuid ();
    gid_t gid = getgid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

Рис. 2: Код программы simpleid.c

3. Скомпилировала программу simpleid, выполнила её и сравнила результат с выводом команды id (рис. 3).

```
[guest@lmponomareva ~]$ nano simpleid.c
[guest@lmponomareva ~]$ gcc simpleid.c -o simpleid
[guest@lmponomareva ~]$ ./simpleid
uid=1001, gid=1001
[guest@lmponomareva ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 3: Установка расширенного атрибута от имени обычного пользователя

Видим совпадение id пользователя и группы.

4. Усложнила программу, добавив вывод действительных идентификаторов (рис. 4).

```
GNU nano 5.6.1                                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_gid=%d\n", real_uid, real_gid);
    return 0;
}
```

Рис. 4: Программа simpleid2.c

5. Скомпилировала и запустила программу simpleid2.c (рис. 5).


```
[guest@lmponomareva ~]$ nano simpleid2.c
[guest@lmponomareva ~]$ gcc simpleid2.c -o simpleid2
[guest@lmponomareva ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@lmponomareva ~]$
```

Рис. 5: Вывод реальных и действительных идентификаторов

6. От имени суперпользователя выполнила команды смены владельца и группы файла simpleid2 и его прав доступа (рис. 6).

```
[guest@lmponomareva ~]$ su
Password:
[root@lmponomareva guest]# chown root:guest /home/guest/simpleid2
[root@lmponomareva guest]# chmod u+s /home/guest/simpleid2
```

Рис. 6: Смена владельца файла и прав доступа

7. Проверила правильность установки новых атрибутов и смены владельца файла simpleid2, запустила программу и сравнила её вывод с результатом ввода команды id: (рис. 7).

```
[guest@lmponomareva ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct  8 11:36 simpleid2
[guest@lmponomareva ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@lmponomareva ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 7: Вывод программы simpleid2

Видим, что действительный идентификатор владельца файла сменился на 0 (root), а реальный не изменился.

8. Проделала тоже самое относительно SetGID-бита. (рис. 8).

```
[guest@lmponomareva ~]$ su -
Password:
[root@lmponomareva ~]# chown root:root /home/guest/simpleid2
[root@lmponomareva ~]# chmod g+s /home/guest/simpleid2
[root@lmponomareva ~]# su guest
[guest@lmponomareva root]$ cd /home/guest
[guest@lmponomareva ~]$ ls -l
total 96
drwx-----. 3 guest guest    31 Oct  1 12:01 dir1
-rwsrwxr-x. 1 root  guest 25952 Oct  8 11:47 readfile
-r------. 1 root  guest  416 Oct  8 11:47 readfile.c
-rwxrwxr-x. 1 guest guest 25904 Oct  8 11:30 simpleid
-rwxrwsr-x. 1 root  root  26008 Oct  8 11:36 simpleid2
-rw-rw-r--. 1 guest guest   310 Oct  8 11:35 simpleid2.c
-rw-rw-r--. 1 guest guest   177 Oct  8 11:30 simpleid.c
[guest@lmponomareva ~]$ ./simpleid2
e_uid=1001, e_gid=0
real_uid=1001, real_gid=1001
```

Рис. 8: SetGID-бит

9. Создала программу readfile.c (рис. 9).

```
GNU nano 5.6.1      readfile.c

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. 9: Код программы readfile.c

10. Откомпилировала её (рис. 10).

```
[guest@lmponomareva ~]$ gcc readfile.c -o readfile
[guest@lmponomareva ~]$
```

Рис. 10: Попытка изменения прав доступа файла

11. Сменила владельца у файла readfile.c (рис. 11).

```
[root@lmponomareva guest]# chown root readfile.c
[root@lmponomareva guest]# ls -l
total 96
drwx----- . 3 guest guest    31 Oct  1 12:01 dir1
-rwxrwxr-x. 1 guest guest 25952 Oct  8 11:47 readfile
-rw-rw-r--. 1 root  guest   416 Oct  8 11:47 readfile.c
```

Рис. 11: Смена владельца файла

12. Изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (рис. 12).

```
[root@lmponomareva guest]# chmod 000 readfile.c
[root@lmponomareva guest]# ls -l
total 96
drwx----- . 3 guest guest    31 Oct  1 12:01 dir1
-rwxrwxr-x. 1 guest guest 25952 Oct  8 11:47 readfile
----- . 1 root  guest   416 Oct  8 11:47 readfile.c
-rwxrwxr-x. 1 guest guest 25904 Oct  8 11:30 simpleid
-rwsrwxr-x. 1 root  guest 26008 Oct  8 11:36 simpleid2
-rw-rw-r--. 1 guest guest   310 Oct  8 11:35 simpleid2.c
-rw-rw-r--. 1 guest guest   177 Oct  8 11:30 simpleid.c
[root@lmponomareva guest]# chmod u+r readfile.c
[root@lmponomareva guest]# ls -l
total 96
drwx----- . 3 guest guest    31 Oct  1 12:01 dir1
-rwxrwxr-x. 1 guest guest 25952 Oct  8 11:47 readfile
-r----- . 1 root  guest   416 Oct  8 11:47 readfile.c
```

Рис. 12: Смена атрибутов

13. Проверила, что пользователь guest не может прочитать файл readfile.c (рис. 13).

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

Рис. 13: Проверка доступа к файлу

14. Сменила у программы readfile владельца и установила SetUID-бит (рис. 14).

```
[root@lmponomareva guest]# chown root readfile
[root@lmponomareva guest]# ls -l
total 96
drwx-----. 3 guest guest    31 Oct  1 12:01 dir1
-rwxrwxr-x. 1 root  guest 25952 Oct  8 11:47 readfile
-r------. 1 root  guest   416 Oct  8 11:47 readfile.c
-rwxrwxr-x. 1 guest guest 25904 Oct  8 11:30 simpleid
-rwsrwxr-x. 1 root  guest 26008 Oct  8 11:36 simpleid2
-rw-rw-r--. 1 guest guest   310 Oct  8 11:35 simpleid2.c
-rw-rw-r--. 1 guest guest   177 Oct  8 11:30 simpleid.c
[root@lmponomareva guest]# chmod u+s readfile
[root@lmponomareva guest]# ls -l
total 96
drwx-----. 3 guest guest    31 Oct  1 12:01 dir1
-rwsrwxr-x. 1 root  guest 25952 Oct  8 11:47 readfile
```

Рис. 14: Смена владельца и добавление SetUID у readfile

15. Проверила, может ли программа readfile прочитать файл readfile.c (рис. 15).

```
[guest@lmponomareva ~]$ ./readfile 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]);
    }
}
```

Рис. 15: Чтение файла readfile.c программой readfile

16. Проверила, может ли программа readfile прочитать файл /etc/shadow (рис. 16).

```
[guest@lmponomareva ~]$ ./readfile /etc/shadow
root:$6$W2IR6nS3QWKTikL2$r4H9pvGoJtHwSwGVHYR/Wv
zjuvHIne08CntHHPH8dqCK1q/::0:99999:7:::
bin:*:19123:0:99999:7:::
daemon:*:19123:0:99999:7:::
adm:*:19123:0:99999:7:::
lp:*:19123:0:99999:7:::
sync:*:19123:0:99999:7:::
shutdown:*:19123:0:99999:7:::
halt:*:19123:0:99999:7:::
mail:*:19123:0:99999:7:::
operator:*:19123:0:99999:7:::
games:*:19123:0:99999:7:::
```

Рис. 16: Чтение файла /etc/shadow программой из readfile

17. Выяснила, установлен ли атрибут Sticky на директории /tmp (рис. 17).

```
[guest@lmponomareva ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Oct  8 21:13 tmp
[guest@lmponomareva ~]$
```

Рис. 17: Атрибут Sticky на директории /tmp

18. От имени пользователя guest создала файл file01.txt в директории /tmp со словом test (рис. 18).

```
[guest@lmponomareva ~]$ echo "test" > /tmp/file01.txt
[guest@lmponomareva ~]$
```

Рис. 18: Создание файла file01.txt в директории /tmp

19. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные» (рис. 19).

```
[guest@lmponomareva ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Oct  8 21:14 /tmp/file01.txt
[guest@lmponomareva ~]$ chmod o+w /tmp/file01.txt
[guest@lmponomareva ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Oct  8 21:14 /tmp/file01.txt
```

Рис. 19: Чтение и запись для категории пользователей «все остальные»

20. От пользователя guest2 (не являющегося владельцем) попробовала прочит-
тать файл /tmp/file01.txt (рис. 20).

```
[guest@lmponomareva ~]$ su guest2
Password:
[guest2@lmponomareva guest]$ cat /tmp/file01.txt
test
```

Рис. 20: Чтение файла /tmp/file01.txt

21. От пользователя guest2 попробовала дозаписать в файл /tmp/file01.txt сло-
во test2 (рис. 21).

```
[guest2@lmponomareva guest]$ echo "test2" >> /tmp/file01.txt
[guest2@lmponomareva guest]$ cat /tmp/file01.txt
test
test2
```

Рис. 21: Дозапись файла /tmp/file01.txt

22. От пользователя guest2 попробовала записать в файл /tmp/file01.txt слово
test3, стерев при этом всю имеющуюся в файле информацию (рис. 22).


```
[guest2@lmponomareva guest]$ echo "test3" > /tmp/file01.txt
[guest2@lmponomareva guest]$ cat /tmp/file01.txt
test3
```

Рис. 22: Перезапись файла /tmp/file01.txt

23. От пользователя guest2 попробовала удалить файл /tmp/file01.txt (рис. 23).

```
[guest2@lmponomareva guest]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@lmponomareva guest]$
```

Рис. 23: Попытка удаления файла /tmp/file01.txt

Файл удалить не удалось.

24. Повысила свои права до суперпользователя следующей командой и сняла атрибут t (Sticky-бит) с директории /tmp (рис. 24).

```
[guest2@lmponomareva guest]$ su -
Password:
[root@lmponomareva ~]# chmod -t /tmp
```

Рис. 24: Снятие Sticky-бит

25. От пользователя guest2 проверила, что атрибута t у директории /tmp нет (рис. 25).

```
[root@lmponomareva ~]# exit
logout
[guest2@lmponomareva guest]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 Oct  8 21:19 tmp
```

Рис. 25: Проверка атрибутов

26. Повторила предыдущие шаги (рис. 26).

```
[guest2@lmponomareva guest]$ cat /tmp/file01.txt
test3
[guest2@lmponomareva guest]$ echo "test4" > /tmp/file01.txt
[guest2@lmponomareva guest]$ cat /tmp/file01.txt
test4
[guest2@lmponomareva guest]$ rm /tmp/file01.txt
[guest2@lmponomareva guest]$
```

Рис. 26: Проверка атрибутов

Теперь можем удалять файлы находящиеся в каталоге tmp.

Выводы

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов.

Список литературы

1. Основы безопасности информационных систем : Учеб. пособие для студентов вузов, обучающихся по специальностям “Компьютер. безопасность” и “Комплекс. обеспечение информ. безопасности автоматизир. систем” / Д.А. Зегжда, А.М. Ивашко. - М. : Горячая линия - Телеком, 2000. - 449, [2] с. : ил., табл.; 21 см.; ISBN 5-93517-018-3.