Отчет по лабораторной работе №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Бурдина Ксения Павловна

2022 Oct 5th

Содержание

List of Figures

1 Цель работы

Целью данной работы является 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 Ход выполнение лабораторной работы

2.1 Создание программы

1. Войдем в систему от имени пользователя guest

2. Создадим программу simpleid.c:

```
[guest@10 ~]$ touch simpleid.c
[guest@10 ~]$ vi simpleid.c
```

Figure 2.1: Создание программы simpleid.c

```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;
}
```

Figure 2.2: Листинг программы simpleid.c

3. Скомпилируем программу и убедимся, что файл программы создан:

```
[guest@10 ~]$ gcc simpleid.c -o simpleid
```

Figure 2.3: Компиляция файла simpleid.c

4. Выполним программу simpleid:

```
[guest@10 ~]$ ./simpleid
uid=1001, gid=1001
```

Figure 2.4: Выполнение программы simpleid

5. Выполним системную программу id:

```
[guest@10 ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfin
ed r:unconfined t:s0-s0:c0.c1023
```

Figure 2.5: Выполнение системной программы id

Видим, что выводимые данные совпадают с данными из прошлого пункта.

6. Усложним 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;
}
```

Figure 2.6: Листинг усложненной программы

Получившуюся пргограмму назовем simpleid2.c.

7. Скомпилируем и запустим simpleid2.c:

```
[guest@10 ~]$ gcc simpleid2.c -o simpleid2
[guest@10 ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Figure 2.7: Компиляция и запуск simpleid2.c

8. От имени суперпользователя выполним команды:

```
[root@10 guest]# chown root:guest /home/guest/simpleid2
[root@10 guest]# chmod u+s /home/guest/simpleid2
```

Figure 2.8: Выполнение команд от имени суперпользователя

9. Для перехода в режим суперпользователя мы выполнили команду su. Если говорить про команду sudo, то ее следует использовать перед каждой командой, если работа производится от имени обычного пользователя, а команды возможны к выполнению только суперпользователем.

10. Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

```
[guest@10 ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct  4 16:44 simpleid2
```

Figure 2.9: Проверка установки атбирутов и смены владельца simpleid2

Видим, что владельцем файла указан root и у пользователя выставлен атрибут s.

11. Запустим simpleid2 и id:

```
[guest@10 ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@10 ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfin
ed r:unconfined t:s0-s0:c0.c1023
```

Figure 2.10: Запуск simpleid2 и id

Видим, что у нас при запуске программы выводятся значения e_uid, e_gid, real_uid и real_gid, которые мы прописали в коде, а при просмотре информации выводятся действительные значения uid и gid.

12. Проделаем то же самое относительно SetGID-бита [1]:

```
[root@10 guest]# chmod g+s /home/guest/simpleid2
```

Figure 2.11: Изменение атрибутов файла для группы пользователей

```
[guest@10 ~]$ ls -l simpleid2
-rwsrwsr-x. 1 root guest 26008 Oct  4 16:44 simpleid2
```

Figure 2.12: Проверка изменения атрибутов

```
[guest@10 ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@10 ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfin
ed r:unconfined t:s0-s0:c0.c1023
```

Figure 2.13: Запуск программы и id

13. Создадим программу readfile.c:

```
[guest@10 ~]$ touch readfile.c
[guest@10 ~]$ nano readfile.c
```

Figure 2.14: Создание программы readfile.c

```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;
}
```

Figure 2.15: Листинг программы readfile.c

14. Откомпилируем программу readfile.c:

```
[guest@10 ~]$ gcc readfile.c -o readfile
```

Figure 2.16: Компиляция программы readfile.c

15. Сменим владельца файла readfile.c и изменим права так, чтоб только суперпользователь мог прочитать его, а guest нет:

```
[root@10 guest]# chown root:guest /home/guest/readfile.c
[root@10 guest]# chmod 730 /home/guest/readfile.c
```

Figure 2.17: Изменение владельца и атрибутов файла readfile.c

16. Проверим, что пользователь guest не может прочитать файл readfile.c:

```
[guest@10 ~]$ ls -l readfile.c
-rwx-wx---. 1 root guest 422 Oct  4 17:30 readfile.c
[guest@10 ~]$ cat readfile.c
cat: readfile.c: Permission denied
```

Figure 2.18: Проверка невозможности чтения файла readfile.c

17. Сменим у программы readfile владельца и установим SetUID-бит:

```
[root@10 guest]# chown root:guest /home/guest/readfile
[root@10 guest]# chmod u+s /home/guest/readfile
```

Figure 2.19: Смена владельца readfile и установка SetUID-бита

18. Проверим, что программа может прочитать файл readfile:

```
[guest@10 ~]$ ./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]);
        }

        while (bytes_read == sizeof (buffer));
        close (fd);
        return 0;
}
```

Figure 2.20: Проверка возможности чтения файла readfile

19. Проверим, что программа readfile может прочитать файл /etc/shadow:

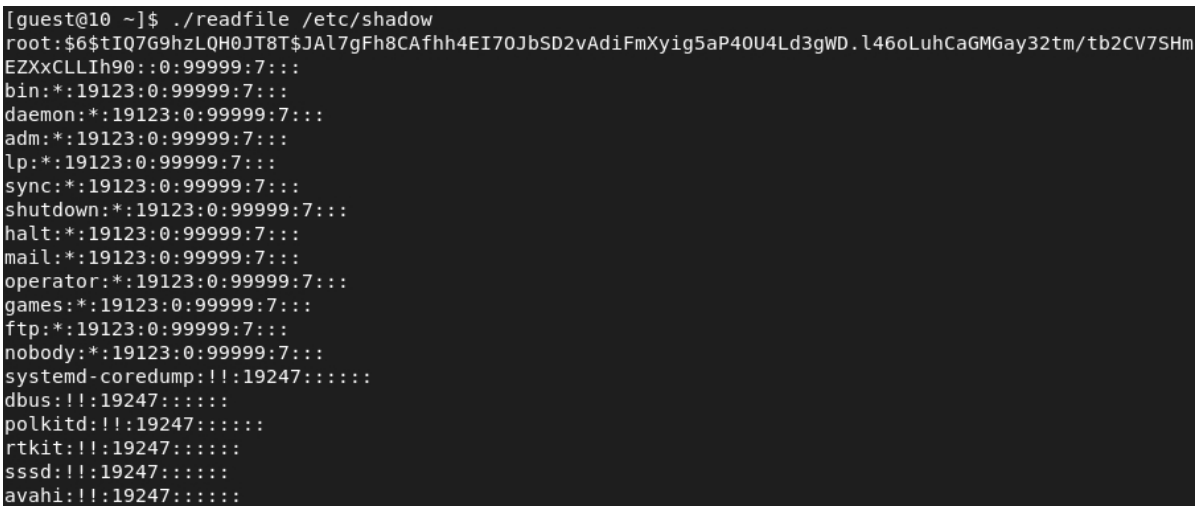

Figure 2.21: Чтение файла /etc/shadow

Можем заметить, что SetUID-бит позволяет пользователю запускать исполняемый файл с правами владельца этого файла.

2.2 Исследование Sticky-бита

1. Выясним, установлен ли атрибут Sticky на директории /tmp:

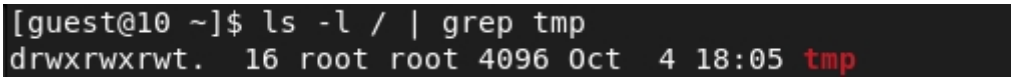

Figure 2.22: Проверка установки атрибута Sticky

2. От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test:

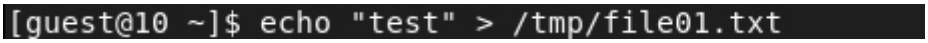

Figure 2.23: Создание файла file01.txt

3. Просмотрим атрибуты у только что созданного файла и расширим чтение и запись для категории пользователей “все остальные”:

```
[guest@10 ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Oct  4 18:11 /tmp/file01.txt
[guest@10 ~]$ chmod o+rw /tmp/file01.txt
[guest@10 ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Oct  4 18:11 /tmp/file01.txt
```

Figure 2.24: Просмотр и изменение атрибутов файла file01.txt

4. От пользователя guest2 (не являющегося владельцем) попробуем прочитать файл /tmp/file01.txt:

```
[guest2@10 guest]$ cat /tmp/file01.txt
test
```

Figure 2.25: Попытка чтения файла от имени guest2

5. От пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2:

```
[guest2@10 guest]$ echo "test2" > /tmp/file01.txt
```

Figure 2.26: Дозапись в файл слова test2

6. Проверим содержимое файла file01.txt:

```
[guest2@10 guest]$ cat /tmp/file01.txt
test2
```

Figure 2.27: Проверка содержимого файла

7. От пользователя guest2 попробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию:

```
[guest2@10 guest]$ echo "test3" > /tmp/file01.txt
```

Figure 2.28: Запись в файл слова test3

8. Проверим содержимое файла:

```
[guest2@10 guest]$ cat /tmp/file01.txt
test3
```

Figure 2.29: Проверка содержимого файла

9. От пользователя guest2 попробуем удалить файл /tmp/file01.txt:

```
[guest2@10 guest]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
```

Figure 2.30: Попытка удаления файла

Мы получили отказ в выполнении действия.

10. Повысим свои права до суперпользователя командой su - и выполним после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp:

```
[guest2@10 guest]$ su -
Password:
[root@10 ~]# chmod -t /tmp
```

Figure 2.31: Снятие атрибута t с директории /tmp

11. Покинем режим суперпользователя:

```
[root@10 ~]# exit
logout
```

Figure 2.32: Покидание режима суперпользователя

12. От пользователя guest2 проверим, что атрибута t у директории /tmp нет:

```
[guest2@10 guest]$ ls -l / | grep tmp
drwxrwxrwx.  16 root root 4096 Oct  4 18:19 tmp
```

Figure 2.33: Проверка отсутствия атрибута t

13. Повторим предыдущие шаги:

```
[guest2@10 guest]$ echo "test4" > /tmp/file01.txt
[guest2@10 guest]$ cat /tmp/file01.txt
test4
[guest2@10 guest]$ rm /tmp/file01.txt
```

Figure 2.34: Выполнение команд со снятым атрибутом t

14. Видим, что теперь мы можем удалить файл file01.txt. Суть в том, что атрибут t как Sticky-бит устанавливает такие права, что файл с данным битом может удалить только тот пользователь, который является владельцем файла [2].

15. Повысим свои права до суперпользователя и вернем атрибут t на директорию /tmp:

```
[guest2@10 guest]$ su -
Password:
[root@10 ~]# chmod +t /tmp
[root@10 ~]# exit
logout
```

Figure 2.35: Возврат атрибута t на директорию

3 Выводы

В ходе работы мы изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов; получили практические навыки работы в консоли с дополнительными атрибутами; рассмотрели работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

4 Список литературы

1. Методические материалы курса [1]
2. Электронный источник “Атрибуты файлов в Linux” [2]