

Шаблон отчёта по лабораторной работе

Лабораторная работа № 5

Мерич Дорук Каймакджыоглу

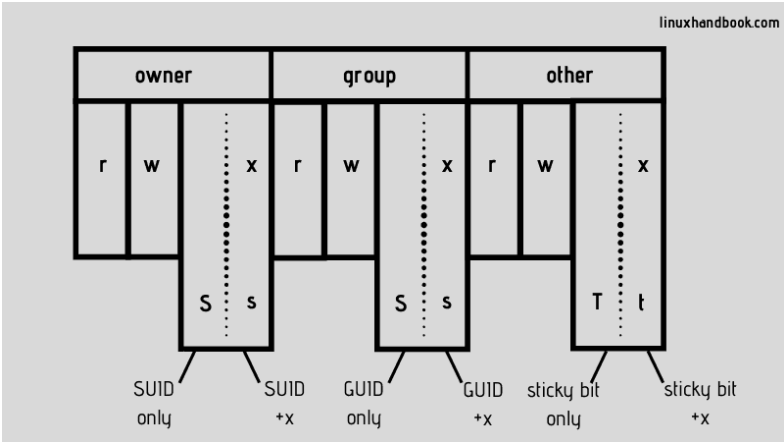
Содержание

Цель работы	1
Теоретическое введение	1
Выполнение лабораторной работы	2
Выводы	6
Список литературы.....	6

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.

Теоретическое введение



В вычислительной технике sticky-бит - это флаг права доступа пользователя, который может быть присвоен файлам и каталогам в Unix-подобных системах. Существует два определения: одно для файлов, другое для каталогов.

Выполнение лабораторной работы

0. Подготовка лабораторного стенда.

```
[root@localhost mericdoruk]# yum install gcc
CentOS Stream 9 - BaseOS                13 kB/s | 9.8 kB    00:00
CentOS Stream 9 - BaseOS                9.4 MB/s | 7.9 MB   00:00
CentOS Stream 9 - AppStream             16 kB/s | 10 kB     00:00
CentOS Stream 9 - AppStream             12 MB/s | 18 MB     00:01
CentOS Stream 9 - Extras packages        42 kB/s | 11 kB     00:00
Package gcc-11.4.1-2.1.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@localhost mericdoruk]# setenforce 0
[root@localhost mericdoruk]# getenforce
Permissive
[root@localhost mericdoruk]# whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/libexec/gcc /usr/share/man/man1/gcc.1.gz /usr/share/info/gcc.info.gz
[root@localhost mericdoruk]# whereis g++
g++: /usr/bin/g++ /usr/share/man/man1/g++.1.gz
```

1. Войдите в систему от имени пользователя guest.
2. создайте программу simpleid.c:
3. Скомпилируйте программу и убедитесь, что файл программы создан:
4. Выполните программу simpleid:
5. Выполните системную программу id: и сравните полученный вами результат с данными предыдущего пункта задания.

```
[guest@localhost Desktop]$ touch simpleid.c
[guest@localhost Desktop]$ vim simpleid.c
[guest@localhost Desktop]$ cat 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;
}
[guest@localhost Desktop]$ gcc simpleid.c -o simpleid
[guest@localhost Desktop]$ ./simpleid
uid=1001, gid=1001
[guest@localhost Desktop]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@localhost Desktop]$
```

6. Усложните программу, добавив вывод действительных идентификаторов:

7. Скомпилируйте и запустите simpleid2.c:

```
[guest@localhost Desktop]$ vim simpleid.c
[guest@localhost Desktop]$ cat simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
    uid_t real_uid = getuid ();
    uid_t uid = geteuid ();

    gid_t real_gid = getgid ();
    gid_t gid = getegid ();

    printf ("uid=%d, gid=%d\n", uid, gid);
    printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
    return 0;
}
[guest@localhost Desktop]$ gcc simpleid.c -o simpleid
[guest@localhost Desktop]$ ./simpleid
uid=1001, gid=1001
real_uid=1001, real_gid=1001
[guest@localhost Desktop]$
```

8. От имени суперпользователя выполните 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`:
11. Запустите `simpleid2` и `id`: Сравните результаты.
12. Прodelайте тоже самое относительно SetGID-бита.

```
[root@localhost Desktop]# chown root:guest simpleid
[root@localhost Desktop]# chmod u+s simpleid
```

```
[guest@localhost Desktop]$ ls -l simpleid
-rwsr-xr-x. 1 root guest 24488 Oct  4 20:08 simpleid
[guest@localhost Desktop]$ ./simpleid
uid=0, gid=1001
real_uid=1001, real_gid=1001
[guest@localhost Desktop]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@localhost Desktop]$
```

13. Создайте программу readfile.c:

```
[guest@localhost Desktop]$ touch readfile.c
[guest@localhost Desktop]$ vim readfile.c
[guest@localhost Desktop]$ cat 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);
}
```

14. Откомпилируйте её.

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.

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?

Отразите полученный результат и ваши объяснения в отчёте.

```
[guest@localhost Desktop]$ vim readfile.c
[guest@localhost Desktop]$ gcc readfile.c -o readfile
[guest@localhost Desktop]$ su
Password:
[root@localhost Desktop]# chown root readfile
[root@localhost Desktop]# pwd
/home/guest/Desktop
[root@localhost Desktop]# sudo su
[root@localhost Desktop]# su guest
[guest@localhost Desktop]$ pwd
/home/guest/Desktop
```

20. Исследование Sticky-бита.

```
[root@localhost mericdoruk]# ls -l / | grep tmp
drwxrwxrwt. 19 root root 4096 Oct 4 20:30 tmp
[root@localhost mericdoruk]#
```

21. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду.

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

23. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»:

```
[root@localhost mericdoruk]# echo "test" > /tmp/file01.txt
[root@localhost mericdoruk]# ls -l /tmp/file01.txt
-rw-r--r--. 1 root root 5 Oct  4 20:31 /tmp/file01.txt
[root@localhost mericdoruk]# chmod o+rw /tmp/file01.txt
[root@localhost mericdoruk]# ls -l /tmp/file01.txt
-rw-r--rw-. 1 root root 5 Oct  4 20:31 /tmp/file01.txt
[root@localhost mericdoruk]#
```

24. т пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt:
25. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой.
26. Проверьте содержимое файла командой.
27. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой:
28. Проверьте содержимое файла командой.
29. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой.

```
[root@localhost mericdoruk]# echo "test2" > /tmp/file01.txt
[root@localhost mericdoruk]# cat /tmp/file01.txt
test2
[root@localhost mericdoruk]# echo "test3" > /tmp/file01.txt
[root@localhost mericdoruk]# cat /tmp/file01.txt
test3
[root@localhost mericdoruk]# rm /tmp/file01.txt
rm: remove regular file '/tmp/file01.txt'? y
[root@localhost mericdoruk]#
```

30. Повысьте свои права до суперпользователя следующей командой su и выполните после этого команду, снимающую атрибут Sticky-бит с директории /tmp:
31. Покиньте режим суперпользователя командой.
32. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет:
33. Повысьте свои права до суперпользователя и верните атрибут t на

```
[root@localhost mericdoruk]# chmod -t /tmp
[root@localhost mericdoruk]# exit
exit
[mericdoruk@localhost ~]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 Oct  4 20:35 tmp
[mericdoruk@localhost ~]$ su
Password:
[root@localhost mericdoruk]# chmod +t /tmp
[root@localhost mericdoruk]#
```

директорию /tmp:

Выводы

Я учил механизмы изменения идентификаторов SetUID- и Sticky-битов. Посмотрел работу механизма смены идентификатора процессов пользователей также влияние бита Sticky на запись и удаление файлов.

Список литературы

Лабораторная работа № 5 {#refs:Лабораторная работа № 5}