Лабораторная работа №5

Информационная безопасность

Дьяконова Софья

Содержание

1 Цель работы 1
2 Теоретическое введение 1
3 Выполнение лабораторной работы 1
3.1 Создание программы 1
3.2 Исследование Sticky-бита 4
4 Выводы 6

1 Цель работы

Целью 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 Теоретическое введение

Setuid, Setgid и Sticky Bit - это специальные типы разрешений позволяют задавать расширенные права доступа на файлы или каталоги.

3 Выполнение лабораторной работы

3.1 Создание программы

Я вошла в систему от имени пользователя guest.(рис. 1)

Далее создала программу simpled.c и заполнила ее.(рис. 1)

Скомплилировала файл через **gcc simpled.c -o simpled** и выполнила программу simpled.(рис. 1)

Выполнила системную программу id. Результаты похожи. Gid и uid одинаковые, однако команда id дает больше информации.(рис. 1)

```
[sadyakonova@localhost ~]$ su - guest
Пароль:
[guest@localhost ~]$ pwd
/home/guest
[guest@localhost ~]$ touch simpled.c
[guest@localhost ~]$ ls
 dir1        Видео       Загрузки      Музыка         'Рабочий стол'
 simpled.c   Документы   Изображения   Общедоступные   Шаблоны
[guest@localhost ~]$ nano simpled.c
[guest@localhost ~]$ cat simpled.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 ~]$ gcc -c simpled.c
[guest@localhost ~]$ gcc -o program simpled.o
[guest@localhost ~]$ ./simpled
-bash: ./simpled: Нет такого файла или каталога
[guest@localhost ~]$ gcc simpled.c -o simpled
[guest@localhost ~]$ ./simpled
uid=1001, gid=1001
[guest@localhost ~]$ nano simpled.c
[guest@localhost ~]$ cat simpled.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
        uid_t real_uid = geteuid ();
        uid_t e_uid = geteuid ();

        gid_t real_gid = getegid ();
        gid_t e_gid = getegid ();
        printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
        printf ("e_uid=%d, e_uid=%d\n", e_uid, e_gid)
        return 0;
}
[guest@localhost ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 1: simpled

Усложнила программу, добавив вывод действительных идентификаторов.(рис. 2)

Скомпилировала и запустила simpled.c.(рис. 2)

От имени суперпользователя выполнила команды **chown root:guest /home/guest/simpled, chmod u+s /home/guest/simpled**.(рис. 2)

Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpled.(рис. 2)

Запустила simpled и id. Результаты похожи. Gid и uid одинаковые, однако команда id дает больше информации.(рис. 2)

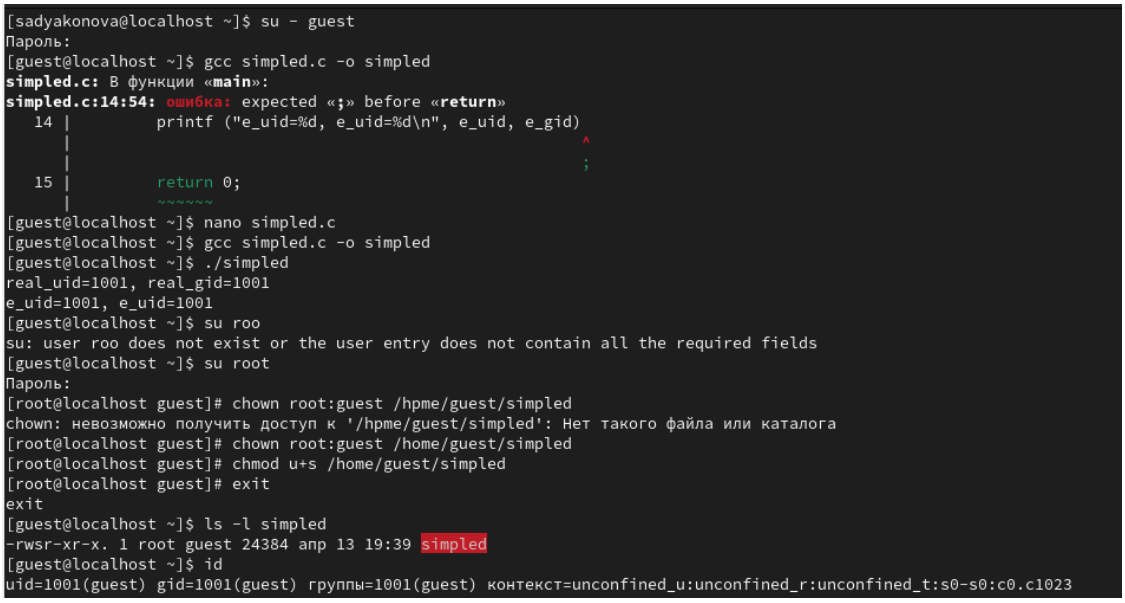

Рис. 2: simpled 2

Создала программу readfile.c.(рис. 3)

Откомпилировала её.(рис. 3)

Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а aleksandrovauv не мог. Проверила, может ли пользователь прочитать файл readfile.c. Не может.(рис. 4)

Программа readfile в целом может прочитать файл /etc/shadow.(рис. 4)

```
[guest@localhost ~]$ touch readfile.c
[guest@localhost ~]$ nano readfile/c
[guest@localhost ~]$ nano readfile.c
[guest@localhost ~]$ 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);
        return 0;
}
[guest@localhost ~]$ gcc redfile.c -o readfile
cc1: фатальная ошибка: redfile.c: Нет такого файла или каталога
компиляция прервана.
[guest@localhost ~]$ gcc readfile.c -o readfile
```

Рис. 3: readfile

```
ароль:
root@localhost guest]# chmod o-r  /home/guest/readfile.c
root@localhost guest]# exit
xit
guest@localhost ~]$ exit
ыход
sadyakonova@localhost ~]$ su - guest
ароль:
guest@localhost ~]$ ls -l readfile.c
rwSr-----. 1 root guest 416 апр 13 19:50 readfile.c
guest@localhost ~]$ cat readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
int
ain (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. 4: readfile

3.2 Исследование Sticky-бита

Выяснила, установлен ли атрибут Sticky на директории /tmp. Установлен.
От имени пользователя guest создала файл file01.txt в директории /tmp со словом

test через **echo “test” > /tmp/file01.txt**. Затем просмотрите атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные» при помощи утилиты **chmod o+rw /tmp/file01.txt**.(рис. 5).

От пользователя guest2 (не являющегося владельцем) попробовала прочитать файл /tmp/file01.txt. Попробовала дозаписать в файл /tmp/file01.txt слово test2 командой **echo “test2” > /tmp/file01.txt**. Мне отказано в доступе. То же самое попробовала сделать с test3, но мне снова отказано в доступе. Файл не записался.(рис. 5).

От пользователя guest2 попробовала удалить файл /tmp/file01.txt. Мне отказали в доступе. Потом я повысила свои права до суперпользователя командой **su -** и сняла атрибут t (Sticky-бит) с директории /tmp. От пользователя guest2 проверила, что атрибута t у директории /tmp нет. Повторила предыдущие шаги. Файл удалился (рис. 5).

Вернула атрибут t (рис. 5).

```
[sadyakonova@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt.  16 root root 4096 апр 13 20:16 tmp
[sadyakonova@localhost ~]$ su - guest
Пароль:
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 апр 13 20:19 /tmp/file01.txt
[guest@localhost ~]$ chmod o+rw /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 апр 13 20:19 /tmp/file01.txt
[guest@localhost ~]$ echo "test2" > /tmp/file01.txt
[guest@localhost ~]$ su - guest2
Пароль:
[guest2@localhost ~]$ echo "test3" > /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@localhost ~]$ echo "test4" > /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@localhost ~]$ su -
Пароль:
[root@localhost ~]# chmod -t /tmp
[root@localhost ~]# exit
выход
[guest2@localhost ~]$ ls -l / | grep tmp
drwxrwxrwx.  16 root root 4096 апр 13 20:23 tmp
[guest2@localhost ~]$ echo "test3" > /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@localhost ~]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
[guest2@localhost ~]$ ls /tmp
systemd-private-37c2b3598df140d5b21f081aadce0876-chronyd.service-PX9kFl
systemd-private-37c2b3598df140d5b21f081aadce0876-colord.service-5XZ7VM
systemd-private-37c2b3598df140d5b21f081aadce0876-dbus-broker.service-MXRIn4
systemd-private-37c2b3598df140d5b21f081aadce0876-kdump.service-OrK6oc
systemd-private-37c2b3598df140d5b21f081aadce0876-ModemManager.service-Bgwsrl
systemd-private-37c2b3598df140d5b21f081aadce0876-power-profiles-daemon.service-Upn9ab
systemd-private-37c2b3598df140d5b21f081aadce0876-rtkit-daemon.service-8dsAJw
systemd-private-37c2b3598df140d5b21f081aadce0876-switcheroo-control.service-vil092
systemd-private-37c2b3598df140d5b21f081aadce0876-systemd-logind.service-uXP3ta
systemd-private-37c2b3598df140d5b21f081aadce0876-upower.service-bq340l
tmpaddon
```

Рис. 5: Sticky

4 Выводы

Я изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Полученила практических навыков работы в консоли с дополнительными атрибутами.