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.

Выполнение лабораторной работы

Первым делом командой `setenforce 0` была отключена система SELinux и командой `getenforce` была проверена корректность выполнения (Рис. 1)

```
[dnbabkov@dnbabkov ~]$ getenforce
Permissive
```

После входа в систему от имени гостевого пользователя была создана программа simpleid (Рис. 2, 3)

```
[guest@dnbabkov ~]$ touch 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;
}
```

Программа была скомпилирована и запущена. После этого было произведено сравнение вывода программы и вывода команды id (Рис. 4):

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

Далее программа была усложнена и сохранена в файл `simpleid2` (Рис. 5)

```
#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;
}
```

После этого команда была скомпилирована и запущена (Рис. 6)

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

От имени суперпользователя была произведена смена пользователя и атрибутов файла `simpleid2`, после чего вновь было запущено выполнение программы и выполнена команда id (Рис. 7, 8, 9):

```
[root@dnbabkov guest]# chown root:guest /home/guest/simpleid2
[root@dnbabkov guest]# chmod u+s /home/guest/simpleid2
-rwsrwxr-x. 1 root  guest 8616 Oct  5 01:08 simpleid2
[guest@dnbabkov ~]$ ./simpleid2
e_uid = 0, e_gid = 1001
real_uid = 1001, real_gid = 1001
[guest@dnbabkov ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfin
ed r:unconfined t:s0-s0:c0.c1023
```

UID отличается по той причине, что у файла другой владелец

Следующем шагом была создана и откомпилированна программа readfile (Рис. 10, 11)

```
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

int main(int argc, char* argv[]) {
        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@dnbabkov ~]$ gcc readfile.c -o readfile
```

Далее были изменены владелец и атрибуты файла `readfile.c` (Рис. 12, 13)

```
[guest@dnbabkov ~]$ su
Password:
[root@dnbabkov guest]# chown root:root /home/guest/readfile.c
[root@dnbabkov guest]# ls -;
ls: cannot access -: No such file or directory
[root@dnbabkov guest]# ls -l
total 48
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Desktop
drwxrwxr-x. 2 guest guest   32 Sep 26 01:57 dir1
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Documents
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Downloads
-rw-rw-r--. 1 guest guest    0 Sep 26 01:40 file1
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Music
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Pictures
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Public
-rwxrwxr-x. 1 guest guest 8552 Oct  5 01:20 readfile
-rw-r--r--. 1 root  root   406 Oct  5 01:20 readfile.c
-rwxrwxr-x. 1 guest guest 8512 Oct  5 01:04 simpleid
-rwsrwxr-x. 1 root  guest 8616 Oct  5 01:08 simpleid2
-rw-r--r--. 1 guest guest  311 Oct  5 01:08 simpleid2.c
-rw-rw-r--. 1 guest guest  311 Oct  5 01:07 simpleid.c
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Templates
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Videos
[root@dnbabkov guest]# chmod 000 readfile.c
[root@dnbabkov guest]# ls -l
total 48
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Desktop
drwxrwxr-x. 2 guest guest   32 Sep 26 01:57 dir1
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Documents
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Downloads
-rw-rw-r--. 1 guest guest    0 Sep 26 01:40 file1
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Music
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Pictures
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Public
-rwxrwxr-x. 1 guest guest 8552 Oct  5 01:20 readfile
----------. 1 root  root   406 Oct  5 01:20 readfile.c
-rwxrwxr-x. 1 guest guest 8512 Oct  5 01:04 simpleid
-rwsrwxr-x. 1 root  guest 8616 Oct  5 01:08 simpleid2
-rw-r--r--. 1 guest guest  311 Oct  5 01:08 simpleid2.c
-rw-rw-r--. 1 guest guest  311 Oct  5 01:07 simpleid.c
drwxr-xr-x. 2 guest guest    6 Oct  5 00:52 Templates
drwxr-xr-x. 2 guest guest    6 Oct _5 00:52 Videos
```

Была произведена проверка возможности пользователя guest прочитать файл `readfile.c` (Рис. 14)

```
[guest@dnbabkov ~]$ cat readfile.c
cat: readfile.c: Permission denied
```

Далее с помощью программы были прочитаны файлы readfile.c и /etc/shadow (Рис. 15, 16)

```
[guest@dnbabkov ~]$ ./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[]) {
        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@dnbabkov ~]$ ./readfile /etc/shadow
root:$6$hLRsmBT1ZyJFKsmB$SqgifOCqSyj8CREVz89g5L8owgZkEV8DQCGPVupvDbuXBgXb/jXV8oH
p61XUE6HD8n63YoXd.P25IXGxZv/Ow1::0:99999:7:::
bin:*:18353:0:99999:7:::
```

Файлы были успешно прочитаны.

Далее командой `ls -l / | grep tmp` было проверено, установлен ли Sticky бит на директории tmp (Рис. 17)

```
[guest@dnbabkov ~]$ ls -l / | grep tmp
drwxrwxrwt.  27 root root 4096 Oct  5 01:28 tmp
```

От имени пользователя был создан файл file01, и добавлены всем пользователям права на чтение и запись в этот файл (Рис. 18)

```
[guest@dnbabkov ~]$ echo "test" > /tmp/file01.txt
[guest@dnbabkov ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Oct  5 01:29 /tmp/file01.txt
[guest@dnbabkov ~]$ chmod o+rw /tmp/file01.txt
[guest@dnbabkov ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Oct  5 01:29 /tmp/file01.txt
```

От имени другого пользователя был прочитан, записан и перезаписан файл file01, а также была произведена неудачная попытка удалить файл (Рис. 19)

```
[guest@dnbabkov ~]$ su guest2
Password:
[guest2@dnbabkov guest]$ cat /tmp/file01.txt
test
[guest2@dnbabkov guest]$ echo "test2" > /tmp/file01.txt
[guest2@dnbabkov guest]$ cat /tmp/file01.txt
test2
[guest2@dnbabkov guest]$ echo "test3" > /tmp/file01.txt
[guest2@dnbabkov guest]$ cat /tmp/file01.txt
test3
[guest2@dnbabkov guest]$ rm /tmp/file01.txt
rm: cannot remove ‘/tmp/file01.txt’: Operation not permitted
```

Далее от имени суперпользователя с директории tmp был снят атрибут t (Рис. 20):

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

Были повторно произведены описанные выше шаги. В этот раз удаление файла было разрешено (Рис. 21)

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

В конце атрибут t был возвращён директории tmp (Рис. 22):

```
[root@dnbabkov guest]# chmod +t /tmp/
```

Вывод

В ходе выполнения лабораторной работы было приобретено понимание принципов работы Sticky- и SetUID-битов