

Презентация по лабораторной работе № 5

Информационная безопасность

Адоле Фейт

06.10.2023

Российский университет дружбы народов, Москва, Россия

Информация

- Адоле Фейт Эне
- студент группы НПМбд-02-20
- Факультет физико-математических и естественных наук
- Российский университет дружбы народов

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.

Создание программы

Сначала созданы и выполнены две программы, имитирующие команду 'id', для отображения различных идентификаторов пользователя и группы. Затем, с правами суперпользователя, установлены биты SetUID и SetGID для этих программ. После этого, при выполнении программы, они получают соответствующие привилегии суперпользователя и группы. Это демонстрирует, как изменение битов SetUID и SetGID может повлиять на выполнение программ и их привилегии.

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

Рис. 1: simpleid.c



```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int
6 main ()
7 {
8     uid_t real_uid = getuid ();
9     uid_t e_uid = geteuid ();
10
11     gid_t real_gid = getgid ();
12     gid_t e_gid = getegid ();
13
14     printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
15     printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
16
17     return 0;
18 }
```

Рис. 2: Код

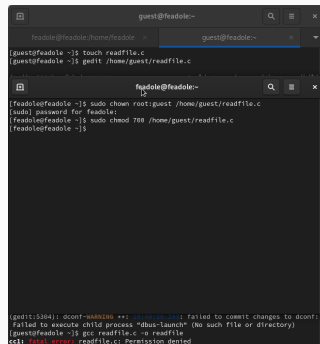
Создание программы(2)

Сначала создали программу для чтения файла (readfile.c) и скомпилировали её. Затем изменили права доступа к программе так, чтобы только пользователь root мог её читать, а гость - нет. Убедились, что гость не имеет доступа к файлу readfile.c через выполнение программы. Далее сменили владельца программы readfile и установили бит SetUID. После этого с помощью программы удалось прочитать файлы readfile.c и /etc/shadow. Этот процесс иллюстрирует изменение прав доступа и привилегий программы в системе.



```
1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <sys/stat.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6
7 int
8 main (int argc, char* argv[])
9 {
10     unsigned char buffer[1024];
11     size_t bytes_read;
12     int fd;
13
14     if (argc < 2)
15         return 1;
16
17     fd = open (argv[1], O_RDONLY);
18     if (fd < 0)
19         return 1;
20
21     while (bytes_read < sizeof (buffer))
22     {
23         bytes_read = read (fd, buffer + bytes_read,
24                           sizeof (buffer) - bytes_read);
25         if (bytes_read < 0)
26             return 1;
27         for (size_t i = 0; i < bytes_read; i++)
28             printf ("%c", buffer[i]);
29     }
30     close (fd);
31     return 0;
32 }
```

Рис. 3: readfile.c



```
guest@feadole:~$ touch readfile.c
guest@feadole:~$ gedit /home/guest/readfile.c

feadole@feadole:~$ sudo chown root:guest /home/guest/readfile.c
[sudo] password for feadole:
feadole@feadole:~$ sudo chmod 700 /home/guest/readfile.c
feadole@feadole:~$

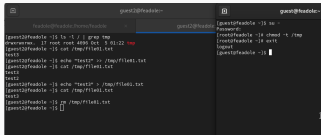
(gedit:5304): dconf-WARNING **: 10:00:00.000: failed to commit changes to dconf:
Failed to execute child process "dbus-launch" (No such file or directory)
(guest@feadole:~$ g++ readfile.c -o readfile
g++: fatal error: readfile.c: Permission denied
```

Исследование Sticky-бита

Сначала мы создали файл в каталоге /tmp, разрешив чтение и запись для всех пользователей. Затем, от имени пользователя guest2, мы попытались прочитать, дозаписать и переписать файл. Однако нам не удалось удалить файл.

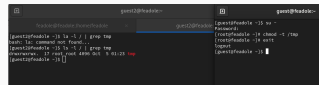
Затем, суперпользователь снял Sticky-бит с каталога tmp и мы повторили действия с файлом. В этот раз удаление файла стало возможным.

Наконец, суперпользователь вернул Sticky-бит на каталог tmp, обеспечивая тем самым ограниченный доступ к файлам в этом каталоге, даже для суперпользователя. Эти действия демонстрируют влияние Sticky-бита на возможности удаления файлов в каталоге.



```
guest2@headon:~$ cd /tmp
guest2@headon:~/tmp$ touch file1.txt
guest2@headon:~/tmp$ ls -la
total 4
drwxrwxrwt 1 root root 4096 Oct 9 01:22 .
drwxrwxrwt 1 root root 4096 Oct 9 01:22 ..
-rw-rw-rw- 1 guest2 guest2 0 Oct 9 01:22 file1.txt
guest2@headon:~/tmp$ cat file1.txt
guest2@headon:~/tmp$ echo "text" > file1.txt
guest2@headon:~/tmp$ cat file1.txt
text
guest2@headon:~/tmp$ echo "text" > file1.txt
guest2@headon:~/tmp$ cat file1.txt
text
guest2@headon:~/tmp$ rm file1.txt
guest2@headon:~/tmp$
```

Рис. 5: sticky-bit(1)



```
guest2@headon:~$ cd /tmp
guest2@headon:~/tmp$ touch file1.txt
guest2@headon:~/tmp$ ls -la
total 4
drwxrwxrwt 1 root root 4096 Oct 9 01:22 .
drwxrwxrwt 1 root root 4096 Oct 9 01:22 ..
-rw-rw-rw- 1 guest2 guest2 0 Oct 9 01:22 file1.txt
guest2@headon:~/tmp$ cat file1.txt
guest2@headon:~/tmp$ echo "text" > file1.txt
guest2@headon:~/tmp$ cat file1.txt
text
guest2@headon:~/tmp$ rm file1.txt
guest2@headon:~/tmp$
```

Рис. 6: sticky-bit(2)

Вывод

В ходе выполнения данной лабораторной работы я изучила механизмы изменения идентификаторов, применение SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.