Защита лабораторной работы №5. Дискреционное разграничения прав в Linux. Исследование влияния дополнительных атрибутов

Бурдина Ксения Павловна

2022 Oct 5th

RUDN University, Moscow, Russian Federation

Результат выполнения лабораторной работы №5

Цель выполнения лабораторной работы

Цель выполнения лабораторной 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
- Влияние бита Sticky на запись и удаление файлов

Результат выполнения лабораторной работы

Результат выполнения лабораторной работы

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 1: Листинг программы simpleid.c

Результат выполнения лабораторной работы

Компиляция программы simpleid.c. Выполнение программ simpleid и id:

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

Figure 2: Компиляция файла

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

Figure 3: Выполнение программы

```
[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 4: Выполнение системной программы

Результат выполнения лабораторной работы

Усложнение программы, добавление вывода действительных идентификаторов:

```c
#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 5: Листинг усложненной программы

Результат выполнения лабораторной работы

Компиляция и запуск 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 6: Компиляция и запуск

Изменение владельца файла и его атрибутов:

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

Figure 7: Смена владельца и атрибутов

Результат выполнения лабораторной работы

Проверка правильности установки новых атрибутов и смены владельца файла simpleid2. Запуск simpleid2 и id:

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

Figure 8: Проверка выполненных действий

```
[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 9: Запуск simpleid2 и id

Результат выполнения лабораторной работы

Действия относительно SetGID-бита:

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

Figure 10: Изменение атрибутов файла для группы пользователей

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

Figure 11: 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 12: Запуск программы и id

Результат выполнения лабораторной работы

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 13: Листинг программы readfile.c

Результат выполнения лабораторной работы

Смена владельца файла readfile.c и изменение прав на чтение:

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

Figure 14: Изменение владельца и атрибутов файла readfile.c

Проверка отсутствия возможности у пользователя guest 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 15: Проверка невозможности чтения файла readfile.c

Результат выполнения лабораторной работы

Компиляция программы readfile.c:

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

Figure 16: Компиляция программы readfile.c

Смена у программы readfile владельца и установка SetUID-бита:

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

Figure 17: Смена владельца readfile и установка SetUID-бита

Результат выполнения лабораторной работы

Проверка чтения файла readfile.c программой 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 18: Проверка возможности чтения файла readfile.c

Результат выполнения лабораторной работы

Проверка чтения файла /etc/shadow программой readfile:

```
[guest@10 ~]$ ./readfile /etc/shadow
root:$6$tIQ7G9hzLQH0JT8T$JAl7gFh8CAfhh4EI7OJbSD2vAdiFmXyig5aP4OU4Ld3gWD.l46oLuhCaGMGay32tm/tb2CV7SHm
EZXxCLLIh90::0:99999:7:::
bin:*:19123:0:99999:7:::
daemon:*:19123:0:99999:7:::
adm:*:19123:0:99999:7:::
lp:*:19123:0:99999:7:::
sync:*:19123:0:99999:7:::
shutdown:*:19123:0:99999:7:::
halt:*:19123:0:99999:7:::
mail:*:19123:0:99999:7:::
operator:*:19123:0:99999:7:::
games:*:19123:0:99999:7:::
ftp:*:19123:0:99999:7:::
nobody:*:19123:0:99999:7:::
systemd-coredump:!!:19247::::::
dbus:!!:19247::::::
polkitd:!!:19247::::::
rtkit:!!:19247::::::
sssd:!!:19247::::::
avahi:!!:19247::::::
```

Figure 19: Чтение файла /etc/shadow

Результат выполнения лабораторной работы

Проверка установки атрибута Sticky на директории /tmp:

```
[guest@10 ~]$ ls -l / | grep tmp
drwxrwxrwt.  16 root root 4096 Oct  4 18:05 tmp
```

Figure 20: Проверка установки Sticky

Создание файла file01.txt в директории /tmp со словом test:

```
[guest@10 ~]$ echo "test" > /tmp/file01.txt
```

Figure 21: Создание файла file01.txt

Просмотр и расширение атрибутов для “остальных” пользователей:

```
[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 22: Просмотр и изменение атрибутов файла file01.txt

Результат выполнения лабораторной работы

Чтение файла от пользователя guest2:

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

Figure 23: Попытка чтения файла от имени guest2

Дозапись в файл от пользователя guest2 слова test2 и проверка содержимого файла:

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

Figure 24: Дозапись в файл слова test2

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

Figure 25: Проверка содержимого файла

Результат выполнения лабораторной работы

Запись в файл /tmp/file01.txt слова test3 и проверка содержимого файла:

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

Figure 26: Запись в файл слова test3

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

Figure 27: Проверка содержимого файла

Попытка удаления файла /tmp/file01.txt:

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

Figure 28: Попытка удаления файла

Результат выполнения лабораторной работы

Повышение прав до суперпользователя и снятие атрибута t (Sticky-бит) с директории /tmp и последующий выход из режима суперпользователя:

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

Figure 29: Снятие атрибута t с директории /tmp

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

Figure 30: Покидание режима суперпользователя

Результат выполнения лабораторной работы

Проверка отсутствия у guest2 атрибута t у директории /tmp:

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

Figure 31: Проверка отсутствия атрибута t

Повтор предыдущих шагов:

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

Figure 32: Выполнение команд со снятым атрибутом t

Выводы

Выводы

1. Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов;
2. Получили практические навыки работы в консоли с дополнительными атрибутами;
3. Рассмотрели работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.