

Лабораторная работа № 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Хамдамова А. А.

12 апреля 2024

Российский университет дружбы народов, Москва, Россия

Информация

- Хамдамова Айжана Абдукаримовна
- студент Факультета Физико-математических и естественных наук
- Российский университет дружбы народов
- 1032225989@pfur.ru
- https://github.com/AizhanaKhamdamova/study_2023-2024_infosec

Вводная часть

1. Дополнительные атрибуты файлов Linux

В Linux существует три основных вида прав — право на чтение (read), запись (write) и выполнение (execute), а также три категории пользователей, к которым они могут применяться — владелец файла (user), группа владельца (group) и все остальные (others). Но, кроме прав чтения, выполнения и записи, есть еще три дополнительных атрибута. [1]

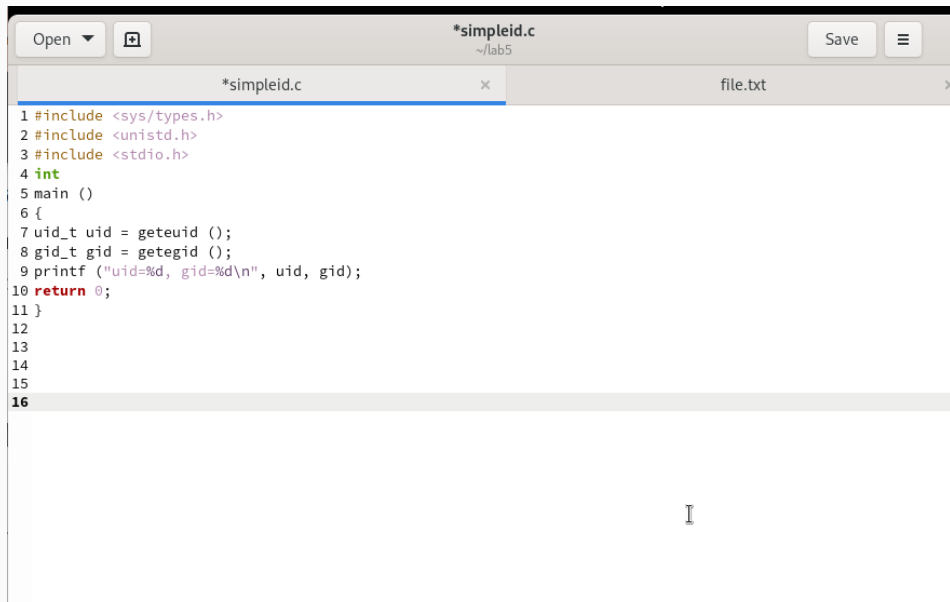
Sticky bit Используется в основном для каталогов, чтобы защитить в них файлы. В такой каталог может писать любой пользователь. Но, из такой директории пользователь может удалить только те файлы, владельцем которых он является. Примером может служить директория /tmp, в которой запись открыта для всех пользователей, но нежелательно удаление чужих файлов.

SUID (Set User ID) Атрибут исполняемого файла, позволяющий запустить его с правами владельца. В Linux приложение запускается с правами пользователя, запустившего указанное приложение. Это обеспечивает дополнительную безопасность т.к. процесс с

- 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

Ход работы

Пример №1



The image shows a code editor window with a title bar containing "Open", a file icon, "*simpleid.c", "~ /lab5", "Save", and a menu icon. Below the title bar, there are two tabs: "*simpleid.c" (active) and "file.txt". The code in the active tab is as follows:

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

A cursor is visible on line 16.

Пример №2



The screenshot shows a code editor window with the title bar 'simpleid2.c' and a path '~ /lab5'. The window contains three tabs: 'simpleid.c', 'file.txt', and 'simpleid2.c'. The 'simpleid2.c' tab is active and displays the following C code:

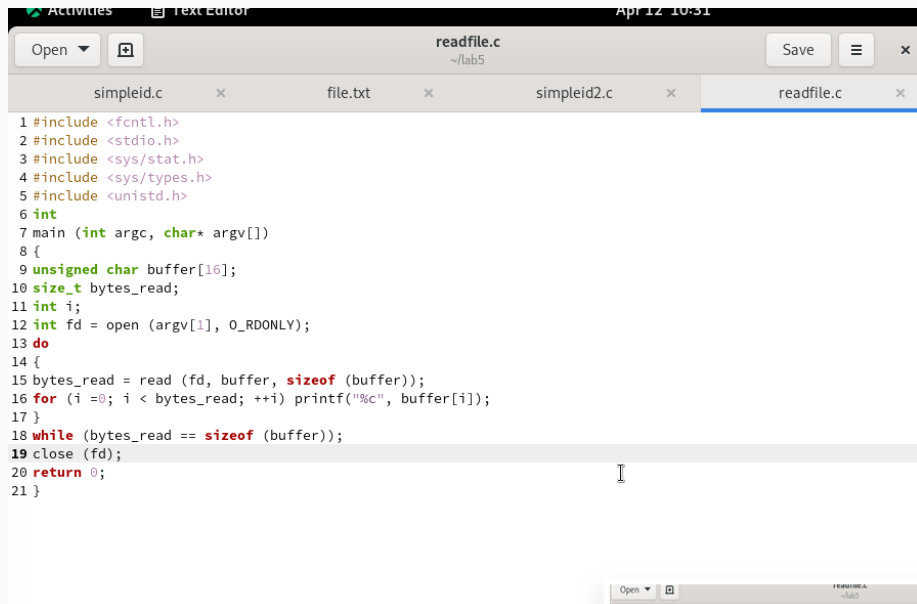
```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4 int
5 main ()
6 {
7     uid_t real_uid = getuid ();
8     uid_t e_uid = geteuid ();
9     gid_t real_gid = getgid ();
10    gid_t e_gid = getegid () ;
11    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
12    printf ("real_uid=%d,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

A cursor is visible on line 15.

Пример №3

```
root@akhamdamova:~  
[root@akhamdamova ~]# ls -l /home/guest/lab5/simpleid2  
-rwsr-xr-x. 1 root guest 26064 Apr 12 10:07 /home/guest/lab5/simpleid2  
[root@akhamdamova ~]# cd /home/guest/lab5/simpleid2  
-bash: cd: /home/guest/lab5/simpleid2: Not a directory  
[root@akhamdamova ~]# /home/guest/lab5/simpleid2  
e_uid=0, e_gid=0  
real_uid=0, real_gid=0  
[root@akhamdamova ~]# /home/guest/lab5/./simpleid2  
e_uid=0, e_gid=0  
real_uid=0, real_gid=0  
[root@akhamdamova ~]# id  
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi  
ned_t:s0-s0:c0.c1023  
[root@akhamdamova ~]# chown root:guest /home/guest/lab5/simpleid2  
[root@akhamdamova ~]# chmod g+s /home/guest/lab5/simpleid2  
[root@akhamdamova ~]# ls -l /home/guest/lab5/simpleid2  
-rwxr-sr-x. 1 root guest 26064 Apr 12 10:07 /home/guest/lab5/simpleid2  
[root@akhamdamova ~]# /home/guest/lab5/./simpleid2  
e_uid=0, e_gid=1001  
real_uid=0, real_gid=0  
[root@akhamdamova ~]# id  
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi  
ned_t:s0-s0:c0.c1023  
[root@akhamdamova ~]#
```

Пример №4



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

Изучила механизмы изменения идентификаторов, применения 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.