

Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Го Чаопэн

7 октября 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Го Чаопэн
- студент НФИбд-02-20
- Российский университет дружбы народов
- 1032194919@pfur.ru
- <https://github.com/LIONUCKY>

Вводная часть

Логические объекты файловой системы (файлы) являются носителями своеобразных меток, которые привычно называют правами доступа. Некоторые метки действительно означают право выполнения определенного действия пользователя над этим объектом. Важно изучить их для дальнейшего применения на практике.

- Атрибуты файлов
- Дистрибутив Rocky
- Дискреционное разграничение доступа

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 работы

От имени пользователя guest создадим программу simpleid.c, скомпилируем ее и убедимся, что файл создан. Выполним команды ./simpleid и id и убедимся, что полученные данные совпадают

```
[guest@guo ~]$ touch simpleid.c
[guest@guo ~]$ nano simpleid.c
[guest@guo ~]$ gcc simpleid.c -o simpleid
[guest@guo ~]$ ./simpleid
uid=1001, gid=1001
[guest@guo ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@guo ~]$
```

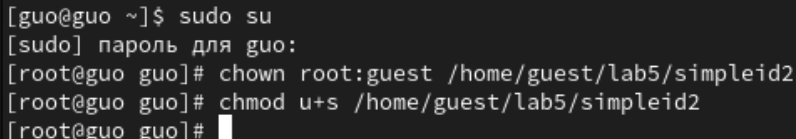
Рис. 1: Использование команд ./simpleid и id

Усложним программу и запишем ее в файл simpleid2.c. Запустим получившуюся программу

```
[guest@guo lab5]$ ./simpleid2  
uid=1001, gid=1001  
[guest@guo lab5]$
```

Рис. 2: Запуск программы simpleid2

От имени суперпользователя установим новые атрибуты и сменим владельца файла simpleid2

A terminal window with a dark background and light gray text. The prompt is [guo@guo ~]\$. The user enters 'sudo su'. The prompt changes to [sudo] пароль для guo:. The user enters a password (not visible). The prompt changes to [root@guo guo]#. The user enters 'chown root:guest /home/guest/lab5/simpleid2'. The prompt changes to [root@guo guo]#. The user enters 'chmod u+s /home/guest/lab5/simpleid2'. The prompt changes to [root@guo guo]#. There is a white cursor at the end of the last line.

```
[guo@guo ~]$ sudo su
[sudo] пароль для guo:
[root@guo guo]# chown root:guest /home/guest/lab5/simpleid2
[root@guo guo]# chmod u+s /home/guest/lab5/simpleid2
[root@guo guo]#
```

Рис. 3: Установки новых атрибутов и смена владельца файла simpleid2

Выполним команду `./simpleid2`

```
[root@guo lab5]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@guo lab5]#
```

Рис. 4: Использование команд `./simpleid2`

Проделаем то же самое относительно SetGID-бита

```
[root@guo lab5]# chmod g+s simpleid2
[root@guo lab5]# ls -l simpleid2
-rwxr-sr-x. 1 root root 26064 окт  7 17:17 simpleid2
[root@guo lab5]# exit
exit
[guest@guo lab5]$ ./simpleid2
e_uid=1001, e_gid=0
real_uid=1001, real_gid=1001
[guest@guo lab5]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@guo lab5]$
```

Рис. 5: Операции с SetGID-битом

Создадим и скомпилируем программу `readfile.c`. Сменим владельца у файла `readfile.c` и изменим права так, чтобы только суперпользователь (`root`) мог прочитать его, а `guest` не мог

```
[guest@guo lab5]$ nano readfile.c
[guest@guo lab5]$ gcc readfile.c -o readfile
[guest@guo lab5]$ su
Пароль:
[root@guo lab5]# chown root:guest readfile.c
[root@guo lab5]# chmod 700 readfile.c
[root@guo lab5]# exit
exit
[guest@guo lab5]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@guo lab5]$
```

Рис. 6: Изменение владельца и прав файла `readfile.c`

Пользователь `guest` не может прочитать файл `readfile.c`

Сменим у программы readfile владельца и установим SetUID-бит

```
[guest@guo lab5]$ su
Пароль:
[root@guo lab5]# chown root:guest readfile
[root@guo lab5]# chmodd u+s readfile
bash: chmodd: command not found...
Similar command is: 'chmod'
[root@guo lab5]# chmod u+s readfile
[root@guo lab5]#
```

Рис. 7: Работа с параметрами readfile

Проверим, может ли программа readfile прочитать файл readfile.c


```
[guest@guo lab5]$ ./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;
}
[guest@guo lab5]$ ./readfile /etc/shadow
```

Рис. 8: Попытка прочитать файл readfile.c программой readfile

Проверим, может ли программа readfile прочитать файл /etc/shadow

```
[guest@guo lab5]$ ./readfile /etc/shadow
root:$6$6Lg6MRXb0EvRPGGeX$Ggp.pI8mf86mlBy0K0f3pWdocBpbxCBvae6yIp1dAItC90Ir76Bd2EBu0TzlefDjeQG9Pmdic5jdDgJV8xJy.:0:99999:7:::
bin:!:19469:0:99999:7:::
daemon:!:19469:0:99999:7:::
adm:!:19469:0:99999:7:::
lp:!:19469:0:99999:7:::
sync:!:19469:0:99999:7:::
shutdown:!:19469:0:99999:7:::
halt:!:19469:0:99999:7:::
mail:!:19469:0:99999:7:::
operator:!:19469:0:99999:7:::
games:!:19469:0:99999:7:::
ftp:!:19469:0:99999:7:::
nobody:!:19469:0:99999:7:::
systemd-coredump:!:19607:!:19607:7:::
dbus:!:19607:!:19607:7:::
polkitd:!:19607:!:19607:7:::
avahi:!:19607:!:19607:7:::
rtkit:!:19607:!:19607:7:::
sssd:!:19607:!:19607:7:::
pipewire:!:19607:!:19607:7:::
libstoragemgmt:!:19607:!:19607:7:::
systemd-oom:!:19607:!:19607:7:::
tss:!:19607:!:19607:7:::
geoclue:!:19607:!:19607:7:::
cockpit-ws:!:19607:!:19607:7:::
cockpit-wsinstance:!:19607:!:19607:7:::
flatpak:!:19607:!:19607:7:::
colord:!:19607:!:19607:7:::
clevi:!:19607:!:19607:7:::
setroubleshoot:!:19607:!:19607:7:::
gdm:!:19607:!:19607:7:::
pesign:!:19607:!:19607:7:::
gnome-initial-setup:!:19607:!:19607:7:::
sshd:!:19607:!:19607:7:::
chrony:!:19607:!:19607:7:::
dnsmasq:!:19607:!:19607:7:::
tcpdump:!:19607:!:19607:7:::
```

Выясним, установлен ли атрибут Sticky на директории /tmp



```
[guest@guo lab5]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 окт 7 17:31 tmp
[guest@guo lab5]$
```

Рис. 10: Чтение атрибутов директории /tmp

От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные»



```
[guest@guo lab5]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 окт  7 17:31 tmp
[guest@guo lab5]$ echo "test" > /tmp/file01.txt
[guest@guo lab5]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 окт  7 17:35 /tmp/file01.txt
[guest@guo lab5]$ chmod o+rw /tmp/file01.txt
[guest@guo lab5]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 окт  7 17:35 /tmp/file01.txt
[guest@guo lab5]$
```


Рис. 11: Создание файла /tmp/file01.txt

От пользователя guest2 попробуем прочитать файл /tmp/file01.txt. Далее попробуем дозаписать в файл /tmp/file01.txt слово test2, записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию. После этого попробуем удалить данный файл

```
[guest2@guo lab5]$ cat /tmp/file01.txt
test
[guest2@guo lab5]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@guo lab5]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@guo lab5]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@guo lab5]$
```

Исследование Sticky-бита

От имени суперпользователя снимем атрибут `t` с директории `/tmp`. От пользователя `guest2` проверим, что атрибута `t` у директории `/tmp` нет. Повторим предыдущие шаги. Теперь мы можем удалить файл



```
[guest2@guo lab5]$ su -
Пароль:
[root@guo ~]# chmod -t /tmp
[root@guo ~]# exit
выход
[guest2@guo lab5]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 окт  7 18:01 tmp
[guest2@guo lab5]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@guo lab5]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@guo lab5]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
[guest2@guo lab5]$
```

В ходе лабораторной работы мне удалось:

- Изучить механизмы изменения идентификаторов, применения SetUID- и Sticky-битов.
- Получить практических навыков работы в консоли с дополнительными атрибутами.
- Рассмотреть работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.