

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Даутов Самир НПИбд-01-19

3 октября, 2022, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи

- SUID - разрешение на установку идентификатора пользователя. Это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла.
- SGID - разрешение на установку идентификатора группы. Принцип работы очень похож на SUID с отличием, что файл будет запускаться пользователем от имени группы, которая владеет файлом.

Цель 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, а также влияние бита Sticky на запись и удаление файлов.

Выполнение лабораторной работы

Программа simpleid

```
[guest@sadautov lab5]$ touch simeid.c
[guest@sadautov lab5]$ mv simeid.c simpleid.c
[guest@sadautov lab5]$ touch simpleie2.c
[guest@sadautov lab5]$ touch readfile.c
[guest@sadautov lab5]$ gedit simpleid.c
[guest@sadautov lab5]$
[guest@sadautov lab5]$
[guest@sadautov lab5]$ gcc simpleid.c
[guest@sadautov lab5]$ gcc simpleid.c -o simpleid
[guest@sadautov lab5]$ ./simpleid
uid=1001, gid=1001
[guest@sadautov lab5]$ id
uid=1001(guest) gid=1001(guest) rpyнны=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@sadautov lab5]$
[guest@sadautov lab5]$
[guest@sadautov lab5]$ gedit
```

Figure 1: результат программы simpleid

Программа simpleid2



```
[guest@sadautov lab5]$ gedit simpleid2.c
[guest@sadautov lab5]$ gcc simpleid2.c
[guest@sadautov lab5]$ gcc simpleid2.c -o simpleid2.c
[guest@sadautov lab5]$ ./simpleid2.c
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@sadautov lab5]$ su
Пароль:
[root@sadautov lab5]# chown root:guest simpleid2.c
[root@sadautov lab5]# chmod u+s simpleid2.c
[root@sadautov lab5]# ./simpleid2.c
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@sadautov lab5]# chmod g+s simpleid2.c
[root@sadautov lab5]# ./simpleid2.c
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@sadautov lab5]# id
uid=0(root) gid=0(root) rгруппы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@sadautov lab5]#
```

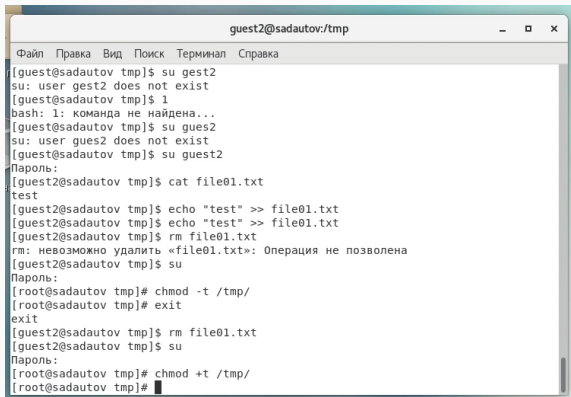
Figure 2: результат программы simpleid2

Программа readfile

```
exit
[guest@sadautov lab5]$
[guest@sadautov lab5]$ gedit readfile.c
[guest@sadautov lab5]$ gcc readfile.c
[guest@sadautov lab5]$ gcc readfile.c -o readfile
[guest@sadautov lab5]$ su
Пароль:
[root@sadautov lab5]# chown root:root readfile
[root@sadautov lab5]# chmod -r readfile.c
[root@sadautov lab5]# chmod u+s readfile
[root@sadautov lab5]# exit
exit
[guest@sadautov lab5]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@sadautov lab5]$ ./readfile readfile.c
```

Figure 3: результат программы readfile

Исследование Sticky-бита



```
guest2@sadautov:/tmp
Файл  Правка  Вид  Поиск  Терминал  Справка
[guest@sadautov tmp]$ su gest2
su: user gest2 does not exist
[guest@sadautov tmp]$ 1
bash: 1: команда не найдена...
[guest@sadautov tmp]$ su gues2
su: user gues2 does not exist
[guest@sadautov tmp]$ su guest2
Пароль:
[guest2@sadautov tmp]$ cat file01.txt
test
[guest2@sadautov tmp]$ echo "test" >> file01.txt
[guest2@sadautov tmp]$ echo "test" >> file01.txt
[guest2@sadautov tmp]$ rm file01.txt
rm: невозможно удалить «file01.txt»: операция не позволена
[guest2@sadautov tmp]$ su
Пароль:
[root@sadautov tmp]# chmod -t /tmp/
[root@sadautov tmp]# exit
exit
[guest2@sadautov tmp]$ rm file01.txt
[guest2@sadautov tmp]$ su
Пароль:
[root@sadautov tmp]# chmod +t /tmp/
[root@sadautov tmp]#
```

Figure 4: исследование Sticky-бита

Выводы

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Также мы рассмотрели работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.