

Лабораторная работа № 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Артамонов Т. Е.

12 сентября 2024

Российский университет дружбы народов, Москва, Россия

Информация

- Артамонов Тимофей Евгеньевич
- студент группы НКНбд-01-21
- Российский университет дружбы народов
- <https://github.com/teartamonov>

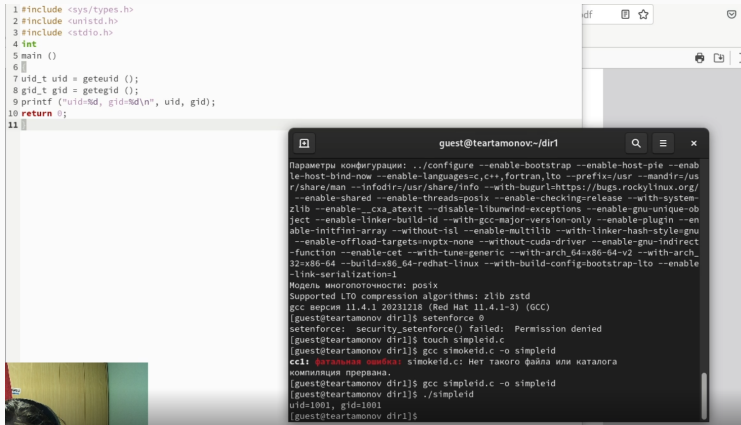


Изучение механизмов изменения идентификаторов, применения SetUID и Sticky-битов.
Получение практических навыков работы в консоли с дополнительными атрибутами.
Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

В настоящее время sticky bit используется в основном для каталогов, чтобы защитить в них файлы. Из такого каталога пользователь может удалить только те файлы, владельцем которых он является. Примером может служить каталог `/tmp`, в который запись открыта для всех пользователей, но нежелательно удаление чужих файлов. Установка атрибута производится утилитой `chmod`.

Выполнение лабораторной работы

Создали файл simple.id и записали в него код из лабораторной. (рис. (fig:001?))



```
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 }
```

```
guest@teartamonov:~/dir1
Параметры конфигурации: ../configure --enable-bootstrap --enable-host-pie --enable-host-bind-now --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=https://bugs.rockylinux.org/ --enable-shared --enable-threads-posix --enable-checking-release --with-system-zlib --enable-_cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-gcc-major-version-only --enable-plugin --enable-initfini-array --without-isl --enable-multilib --with-linker-hash-style=gnu --enable-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-function --enable-cet --with-tune=generic --with-arch_64=x86-64-v2 --with-arch_32=x86-64 --build=x86_64-redhat-linux --with-build-config=bootstrap-lto --enable-link-serialization=1
Модель многопоточности: posix
Supported LTO compression algorithms: zlib zstd
gcc версия 11.4.1 20231218 (Red Hat 11.4.1-3) (GCC)
[guest@teartamonov dir1]$ setenforce 0
setenforce: security_setenforce() failed: Permission denied
[guest@teartamonov dir1]$ touch simpleid.c
[guest@teartamonov dir1]$ gcc smokeid.c -o simpleid
cc1: фатальная ошибка: smokeid.c: Нет такого файла или каталога
компиляция прервана.
[guest@teartamonov dir1]$ gcc simpleid.c -o simpleid
[guest@teartamonov dir1]$ ./simpleid
uid=1001, gid=1001
[guest@teartamonov dir1]$
```

Рис. 1: После запуска получили uid и gid нашего пользователя

Усложним скрипт, добавив вывод real uid и gid. (рис. (fig:002?))

```
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, real_gid);
13    return 0;
14 }
```

```
guest@teartamonov:~/dir1
[guest@teartamonov dir1]$ id
uid=1001(guest) gid=1001(guest) rpynm=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@teartamonov dir1]$ gcc simpleid.c -o simpleid
[guest@teartamonov dir1]$ id
uid=1001(guest) gid=1001(guest) rpynm=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@teartamonov dir1]$ ./simpleid
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@teartamonov dir1]$ gcc simpleid.c -o simpleid
[guest@teartamonov dir1]$ gcc simpleid2.c -o simpleid2
simpleid2.c: 8 функция «main»:
simpleid2.c:13:11: warning: в программе обнаружен некорректный символ «\342»
13 | real_gid);→
   |             ^
simpleid2.c:13:12: warning: в программе обнаружен некорректный символ «\342»
13 | real_gid);→
   |             ^
[guest@teartamonov dir1]$ gcc simpleid2.c -o simpleid2
[guest@teartamonov dir1]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@teartamonov dir1]$
```

Рис. 2: Теперь выводятся и real uid и gid, все совпадает с результатами предыдущих шагов

Пропишем chown и chmod. (рис. (fig:003?))

```
[root@teartamonov ~]# chown root:guest /home/guest/dir1/simpleid2
[root@teartamonov ~]# chmod u+s /home/guest/dir1/simpleid2
bash: chmod: команда не найдена...
Аналогичная команда: 'chmod'
[root@teartamonov ~]# chmod u+s /home/guest/dir1/simpleid2
[root@teartamonov ~]#
```

```
[guest@teartamonov dir1]$ ls -l
итого 52
-rw-----. 1 guest guest 10 сен 8 00:25 file1
-rwxr-xr-x. 1 guest guest 17616 сен 8 00:50 simpleid
-rwsr-xr-x. 1 root guest 17720 сен 8 00:51 simpleid2
-rw-r--r--. 1 root root 302 сен 8 00:50 simpleid2.c
-rw-r--r--. 1 guest guest 175 сен 8 00:50 simpleid.c
[guest@teartamonov dir1]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
```

Рис. 3: chown изменяет владельца файла, а chmod u+s позволяет запускать файл с правами владельца. Теперь при запуске файла от имени guest получаем e_uid root

Проделаем то же самое с SetGID-битом. (рис. (fig:004?))

```
[root@teartamonov dir1]# chmod u-s /home/guest/dir1/simpleid2
[root@teartamonov dir1]# chmod g+s /home/guest/dir1/simpleid2
[root@teartamonov dir1]#
```

```
[guest@teartamonov dir1]$ ls -l
total 52
-rw-r--r--. 1 guest guest 10 сен 8 00:25 file1
-rwxr-xr-x. 1 guest guest 17616 сен 8 00:50 simpleid
-rwxr-sr-x. 1 root guest 17720 сен 8 00:51 simpleid2
-rw-r--r--. 1 root root 302 сен 8 00:50 simpleid3
-rw-r--r--. 1 guest guest 175 сен 8 00:50 simpleid4
[guest@teartamonov dir1]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@teartamonov dir1]$ id
uid=1001(guest) gid=1001(guest) rpyнны=1001(guest)
nеd_r:unconfined_t:s0-s0:c0.c1023
[guest@teartamonov dir1]$
```

Получившую

7. Скомпилируй gcc simpleid2.c -o simpleid2
8. От имени су

Рис. 4: Вывод такой же

Создадим файл readfile.c как в лабораторной и скомпилируем его. (рис. (fig:005?))

```
[root@teartamonov dir1]# chown root:guest /home/guest/dir1/readfile.c  
[root@teartamonov dir1]# chown root:guest /home/guest/dir1/readfile.c  
[root@teartamonov dir1]# chmod 700 /home/guest/dir1/simpleid2  
-rw-r--r--. 1 gue
```

Рис. 5: Меняем владельца на root и забираем все права у всех кроме владельца

```
[guest@teartamonov dir1]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@teartamonov dir1]$
```

Рис. 6: guest не может прочесть readfile.c

Попробуем прочитать readfile.c с помощью readfile. (рис. (fig:007?))

```
[guest@teartamonov dir1]$ ./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@teartamonov dir1]$ ./readfile /etc/shadow
```

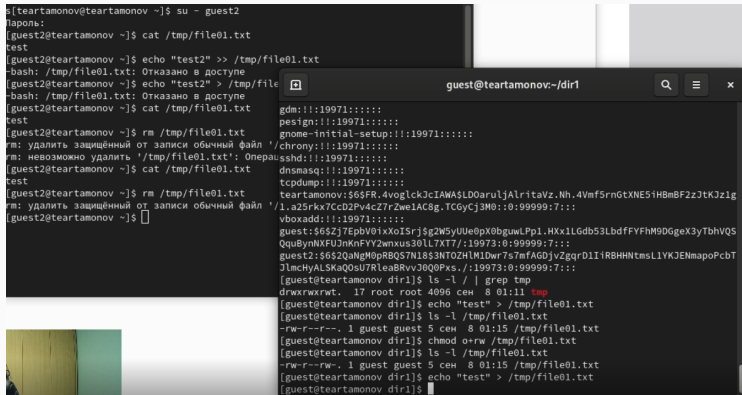
Рис. 7: Успешно

Попробуем прочитать /etc/shadow с помощью readfile. (рис. (fig:008?))

```
cockpit-ws:!!:19971::::::
cockpit-wsinstance:!!:19971::::::
rtkit:!!:19971::::::
pipewire:!!:19971::::::
libstorageengine:!!:19971::::::
flatpak:!!:19971::::::
colord:!!:19971::::::
clevvis:!!:19971::::::
setroubleshoot:!!:19971::::::
gdm:!!:19971::::::
pesign:!!:19971::::::
gnome-initial-setup:!!:19971::::::
chrony:!!:19971::::::
sshd:!!:19971::::::
dnsmasq:!!:19971::::::
tcpdump:!!:19971::::::
teartamonov:$6$FR.4voglckJcIAWA$LD0aruljAlritaVz.Nh.4VmF5rnGtXNE5iHBmBF2zJtKJzlg
1.a25rKx7CcD2Pv4cZ7rZwe1AC8g.TCGyCj3M0::0:99999:7::
vboxadd:!!:19971::::::
guest:$6$Zj7EpbV0ixXoISrj$g2W5yUue0pX0bguwLPp1.HXx1LGdb53LbdfFYFhM9DGgeX3yTbhVQS
QquBynNXFUJnKnFYy2wnxus30lL7XT7/:19973:0:99999:7::
guest2:$6$2QaNgM0pRBQS7N18$3NTOZHlM1Dwr7s7mfAGDjvZgqrD1IiRBHhNtmsL1YKJENmapoPcbT
JlmcHyALSKaQOsU7RleaBRvvJ0Q0Pxs./:19973:0:99999:7::
[guest@teartamonov dir1]$
```

Рис. 8: Успешно

Найдем директорию tmp, создадим там файл от имени guest и от имени guest2 попробуем выполнить с ним разные действия. (рис. (fig:009?))



```
[teartamonov@teartamonov ~]$ su - guest2
Пользователь:
[guest2@teartamonov ~]$ cat /tmp/file01.txt
test
[guest2@teartamonov ~]$ echo "test2" >> /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@teartamonov ~]$ echo "test2" > /tmp/file01.txt
-bash: /tmp/file01.txt: Отказано в доступе
[guest2@teartamonov ~]$ cat /tmp/file01.txt
test
[guest2@teartamonov ~]$ rm /tmp/file01.txt
rm: удалить защищенный от записи обычный файл '/tmp/file01.txt': 0 не удалось
[guest2@teartamonov ~]$ cat /tmp/file01.txt
test
[guest2@teartamonov ~]$ rm /tmp/file01.txt
rm: удалить защищенный от записи обычный файл '/tmp/file01.txt': 0 не удалось
[guest2@teartamonov ~]$

[teartamonov@teartamonov ~]$ su - guest
guest@teartamonov:~/dir1$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 сен  8 01:11 tmp
[guest@teartamonov dir1]$ echo "test" > /tmp/file01.txt
[guest@teartamonov dir1]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 сен  8 01:15 /tmp/file01.txt
[guest@teartamonov dir1]$ chmod o+rw /tmp/file01.txt
[guest@teartamonov dir1]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 сен  8 01:15 /tmp/file01.txt
[guest@teartamonov dir1]$ echo "test" > /tmp/file01.txt
[guest@teartamonov dir1]$
```

Рис. 9: Можем только читать файл, все что связано с изменением запрещено

Уберем параметр -t и попробуем еще раз. (рис. (fig:010?))

```
[root@teartamonov dir1]# chmod -t /tmp
[root@teartamonov dir1]#
[guest2@teartamonov ~]$ rm /tmp/file01.txt
rm: удалить защищенный от записи обычный файл '/tmp/file01.txt'? y
[guest2@teartamonov ~]$
```

Рис. 10: Теперь изменение не для владельца открыто

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

1. Stickybit [Электронный ресурс]. Wikimedia Foundation, Inc., 2024. URL: https://ru.wikipedia.org/wiki/Sticky_bit.