Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Топонен Никита Андреевич

Содержание

Список иллюстраций

Список таблиц

Цель 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.

Задание

Выполнить задания из лабораторной работы и проанализировать полученные результаты.

Теоретическое введение

Для выполнения данной лабораторной нет специальной теории. Необходимы общие знания в области компьютерных наук.

Выполнение лабораторной работы

Проверю, установлен ли у меня компилятор gcc командой gcc –v. У меня он уже установлен как видно на рисунке ниже.

```
[guest@natoponen ~]$ gcc -v
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --prefix=/usr --mandir=/usr/share/man --inf
odir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-
bootstrap --enable-shared --enable-threads=posix --enable-checking=release --wit
h-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-
unique-object --enable-linker-build-id --with-linker-hash-style=gnu --enable-lan
guages=c,c++,objc,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-initf
ini-array --disable-libgcj --with-isl=/builddir/build/BUILD/gcc-4.8.5-20150702/o
bj-x86_64-redhat-linux/isl-install --with-cloog=/builddir/build/BUILD/gcc-4.8.5-
20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --
with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux
Модель многопоточности: posix
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)
```

Рис. 1: Компилятор gcc

Установил setenforce в 0 и провел, что данная команда выполнилась

```
[guest@natoponen ~]$ su
Пароль:
[root@natoponen guest]# setenforce 0
[root@natoponen guest]# getenforce
Permissive
```

Рис. 2: Отключение SELinux

Вошел в систему от имени пользователя guest и создал программу simpleid.c

```
[guest@natoponen ~]$ touch simpleid.c
[guest@natoponen ~]$ ls
dir1        Видео      Загрузки     Музыка          Рабочий стол
simpleid.c  Документы  Изображения  Общедоступные   Шаблоны
[guest@natoponen ~]$ vim simpleid.c
```

Рис. 3: Отключение SELinux

```
#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;
}
```

Скомпилирую программу командой *gcc simpleid.**c** -o simpleid* и запустил ее

```
[guest@natoponen ~]$ gcc simpleid.c -o simpleid
[guest@natoponen ~]$ ./simpleid
uid=1001, gid=1001
```

Рис. 4: Компиляция и выполнение simpleid.c

Выполню системную программу id командой *id*. Результат совпадает

```
[guest@natoponen ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 5: Результат команды id

Усложню программу, добавив вывод 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;
}
```

Скомпилирую в новый файл simpleid2

```
[guest@natoponen ~]$ gcc simpleid.c -o simpleid2
[guest@natoponen ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Рис. 6: Компиляция и выполнение simpleid2

От имени суперпользователя выполню команды:

1. *chown root:guest /home/guest/simpleid2**
2. **chmod u+s /home/guest/simpleid2*

```
[guest@natoponen ~]$ su
Пароль:
[root@natoponen guest]# chown root:guest /home/guest/simpleid2
[root@natoponen guest]# chmod u+s /home/guest/simpleid2
```

Рис. 7: Изменение владельца и прав на файл simpleid2

Команда *chown root:guest /home/guest/simpleid2* меняет владельца файла. Команда *chmod u+**s** /home/**guest**/simpleid2* меняет права доступа к файлу.

Проверю правильность установки новых атрибутов и смены владельца файла simpleid2 командой: *ls -l simpleid2*

```
[root@natoponen guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8616 окт  3 22:43 simpleid2
```

Рис. 8: Атрибуты и владелец файла simpleid2

Запущу simpleid2 и id, команды: *./simpleid2* и *id*

```
[root@natoponen guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@natoponen guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconf
ined_t:s0-s0:c0.c1023
```

Рис. 9: Выполнение simpleid2 и id

Создам программу 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@natoponen ~]$ touch readfile.c
[guest@natoponen ~]$ vim readfile.c
```

Рис. 10: Программа readfile.c

Скомпилирую её командой: *gcc readfile.c -o readfile*

Сменю владельца у файла readfile.c и изменю права так, чтобы только супер-пользователь (root) мог прочитать его, а guest не мог

```
[guest@natoponen ~]$ gcc readfile.c -o readfile
[guest@natoponen ~]$ chown root:guest readfile.c
chown: изменение владельца «readfile.c»: Операция не позволена
[guest@natoponen ~]$ su
Пароль:
[root@natoponen guest]# chown root:guest readfile.c
[root@natoponen guest]# chmod 700 readfile.c
[root@natoponen guest]# su guest
[guest@natoponen ~]$ vim readfile.c
```

Рис. 11: Изменение владельца и прав на файл readfile.c

Проверю, что пользователь guest не может прочитать файл readfile.c.

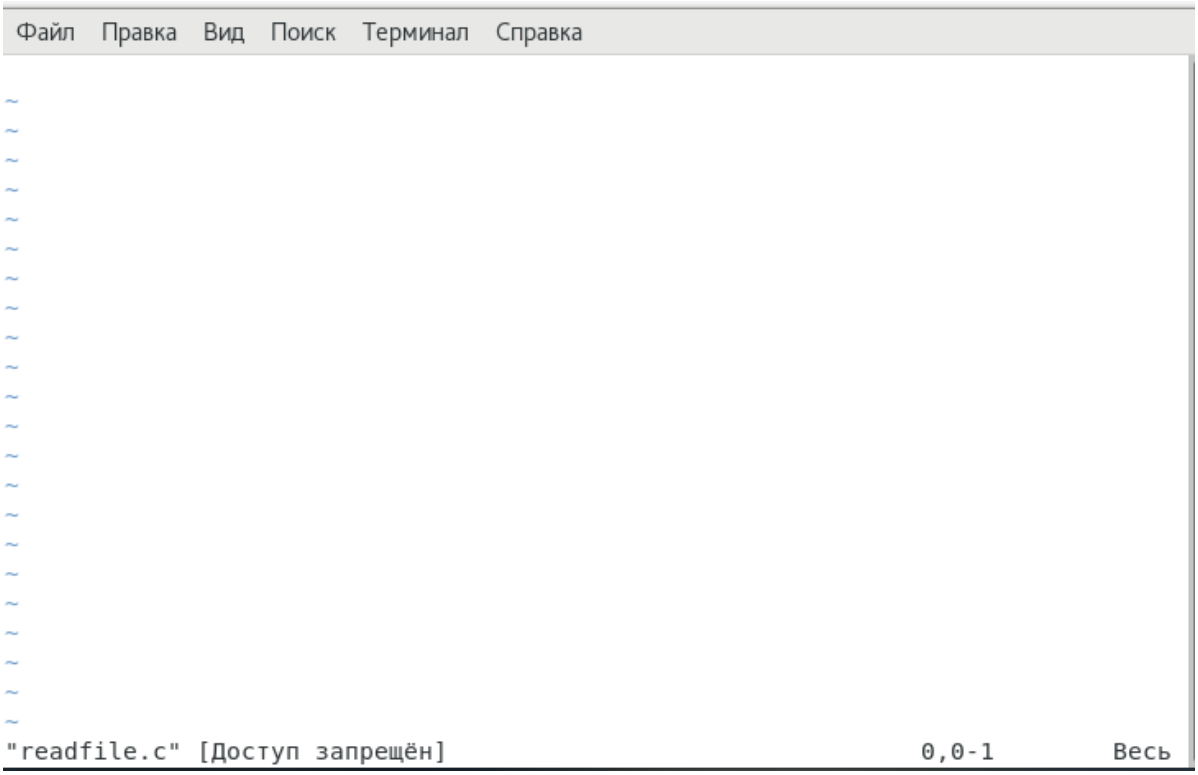

Рис. 12: Отказ в чтении пользователю guest

Сменю у программы readfile владельца и установлю SetUID-бит

```
[guest@natoponen ~]$ su
Пароль:
[root@natoponen guest]# chown root:guest readfile
[root@natoponen guest]# chmod u+s readfile
```

Рис. 13: Установка UID бита для readfile.c

Проверю, может ли программа readfile прочитать файл readfile.c

```
[root@natoponen guest]# ./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;
}
```

Рис. 14: Выполнение программы для файла readfile.c

```
[root@natoponen guest]# ./readfile /etc/shadow
root:$6$TGptGmRXmrZUPQ.V$TGIlQ7bnDoQ6k74EbOVDTzYTFlyxG8ajLtABObjiu5GmN38EzlZ0aK4
ChUisX9F7H4reBFVUb3mRXn0zF0w0o0::0:99999:7:::
bin:*:18353:0:99999:7:::
daemon:*:18353:0:99999:7:::
adm:*:18353:0:99999:7:::
lp:*:18353:0:99999:7:::
sync:*:18353:0:99999:7:::
shutdown:*:18353:0:99999:7:::
halt:*:18353:0:99999:7:::
mail:*:18353:0:99999:7:::
operator:*:18353:0:99999:7:::
games:*:18353:0:99999:7:::
ftp:*:18353:0:99999:7:::
nobody:*:18353:0:99999:7:::
systemd-network:!!:19241::::::
dbus:!!:19241::::::
polkitd:!!:19241::::::
libstoragemgmt:!!:19241::::::
colord:!!:19241::::::
rpc:!!:19241:0:99999:7:::
saned:!!:19241::::::
```

Рис. 15: Выполнение программы для файла /etc/shadow

Поскольку у программы установлен SetUID-бит, то ей временно предоставля-

ются права владельца файла (суперпользователя). Поэтому программа может прочитать файл с правами доступа только для владельца суперпользователя

Выясню, установлен ли атрибут Sticky на директории /tmp, для чего выполню команду

ls -l / | grep tmp

От имени пользователя guest создам файл file01.txt в директории /tmp со словом test: *echo "test" > /tmp/file01.txt*

Просмотрю атрибуты у только что созданного файла и разрешу чтение и запись для категории пользователей «все остальные»:

ls -l /tmp/file01.txt

chmod o+rw /tmp/file01.txt

ls -l /tmp/file01.txt

От пользователя guest2 (не являющегося владельцем) попробую прочитать файл /tmp/file01.txt: *cat /**tmp**/file01.**txt**

От пользователя guest2 попробую дозаписать в файл /tmp/file01.txt слово test2 командой *echo "test2" » /tmp/file01.txt*. Мне удалось выполнить операцию.

Проверю содержимое файла командой *cat /tmp/file01.txt*

От пользователя guest2 попробую записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo *"test3" > /tmp/file01.txt*. Мне удалось выполнить операцию.

Проверю содержимое файла командой *cat /tmp/file01.txt*

От пользователя guest2 попробую удалить файл /tmp/file01.txt командой *rm /tmp/file01.txt*. Мне не удалось удалить файл

```
[guest@natoponen ~]$ ls -l / | grep tmp
drwxrwxrwt.  15 root root 4096 окт  3 23:01 tmp
[guest@natoponen ~]$ echo "test" > /tmp/file01.txt
[guest@natoponen ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт  3 23:04 /tmp/file01.txt
[guest@natoponen ~]$ chmod o+rw /tmp/file01.txt
[guest@natoponen ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 окт  3 23:04 /tmp/file01.txt
[guest@natoponen ~]$ su guest2
Пароль:
[guest2@natoponen guest]$ cat /tmp/file01.txt
test
[guest2@natoponen guest]$ echo "test2" > /tmp/file01.txt
[guest2@natoponen guest]$ cat /tmp/file01.txt
test2
[guest2@natoponen guest]$ echo "test3" > /tmp/file01.txt
[guest2@natoponen guest]$ cat /tmp/file01.txt
test3
[guest2@natoponen guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
```

Рис. 16: Работа со Sticky битом

Повышу свои права до суперпользователя следующей командой *su* и выполню после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: *chmod -t /tmp*

От пользователя guest2 проверил, что атрибута t у директории /tmp нет: *ls -l / | grep tmp*

Повторю предыдущие шаги

```
[root@natoponen guest]# chmod -t /tmp/
[root@natoponen guest]# su guest2
[guest2@natoponen guest]$ ls -l / | grep tmp
drwxrwxrwx.  15 root root 4096 окт  3 23:07 tmp
[guest2@natoponen guest]$ echo "test4" > /tmp/file01.txt
[guest2@natoponen guest]$ cat /tmp/file01.txt
test4
[guest2@natoponen guest]$ rm /tmp/file01.txt
[guest2@natoponen guest]$ cd /tmp/
[guest2@natoponen tmp]$ ls
ssh-VRF8jE9qROYD
systemd-private-f1ab0ea4e2b94e828873299a217c90c2-bolt.service-1qpHGa
systemd-private-f1ab0ea4e2b94e828873299a217c90c2-colord.service-xcczcG
systemd-private-f1ab0ea4e2b94e828873299a217c90c2-cups.service-wNJwa2
systemd-private-f1ab0ea4e2b94e828873299a217c90c2-fwupd.service-UWMFVk
systemd-private-f1ab0ea4e2b94e828873299a217c90c2-rtkit-daemon.service-TvmO2P
tracker-extract-files.1001
yum_save_tx.2022-09-25.17-03.WeLvQ6.yumtx
yum_save_tx.2022-10-03.22-34.R5HCjr.yumtx
```

Рис. 17: Работа с файлом без Sticly бита

Мне удалось удалить файл от имени пользователя, не являющегося его владельцем. Это связано с тем, что Sticky-bit позволяет защищать файлы от случайного удаления, когда несколько пользователей имеют права на запись в один и тот же каталог. Если у файла атрибут t стоит, значит пользователь может удалить файл, только если он является пользователем-владельцем файла или каталога, в котором содержится файл. Если же этот атрибут не установлен, то удалить файл могут все пользователи, которым позволено удалять файлы из каталога.

Повышу свои права до суперпользователя и верну атрибут t на директорию /tmp:

su

chmod +t /tmp

exit

```
[guest2@natoponen tmp]$ su
Пароль:
[root@natoponen tmp]# chmod +t /tmp/
[root@natoponen tmp]# exit
exit
```

Рис. 18: Установление атрибута t

Выводы

В ходе данной лабораторной работы я изучил механизмы изменения идентификаторов, применения SetUID-, SetGID- и Sticky-битов. Рассмотрел работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Список литературы

- Кулябов Д. С., Королькова А. В., Геворкян М. Н Лабораторная работа №5