Отчёт по лабораторной работе №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Ким Реачна

Содержание

Список иллюстраций

1 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.

2 Выполнение лабораторной работы

2.1 Подготовка

1. Для выполнения части заданий требуются средства разработки приложений. Проверили наличие установленного компилятора gcc командой `gcc -v`: компилятор обнаружен.

2. Чтобы система защиты SELinux не мешала выполнению заданий работы, отключили систему запретов до очередной перезагрузки системы командой setenforce 0:

3. Команда getenforce вывела Permissive:

```
[guest@kreachna ~]$ su
Password:
[root@kreachna guest]# gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-redhat-linux
Configured with: ../configure --enable-bootstrap --enable-host-pie --enable-host-bind-n
ow --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-th
reads=posix --enable-checking=release --with-system-zlib --enable-__cxa_atexit --disabl
e-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-gcc-m
ajor-version-only --enable-plugin --enable-initfini-array --without-isl --enable-multil
ib --with-linker-hash-style=gnu --enable-offload-targets=nvptx-none --without-cuda-driv
er --enable-gnu-indirect-function --enable-cet --with-tune=generic --with-arch_64=x86-6
4-v2 --with-arch_32=x86-64 --build=x86_64-redhat-linux --with-build-config=bootstrap-lt
o --enable-link-serialization=1
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.3.1 20221121 (Red Hat 11.3.1-4) (GCC)
[root@kreachna guest]# setenforce 0
[root@kreachna guest]# getenforce
Permissive
[root@kreachna guest]# exit
exit
[guest@kreachna ~]$
```

Рис. 2.1: Подготовка к работе

2.2 Изучение механики SetUID

1. Вошли в систему от имени пользователя guest.

2. Написали программу simpleid.c.

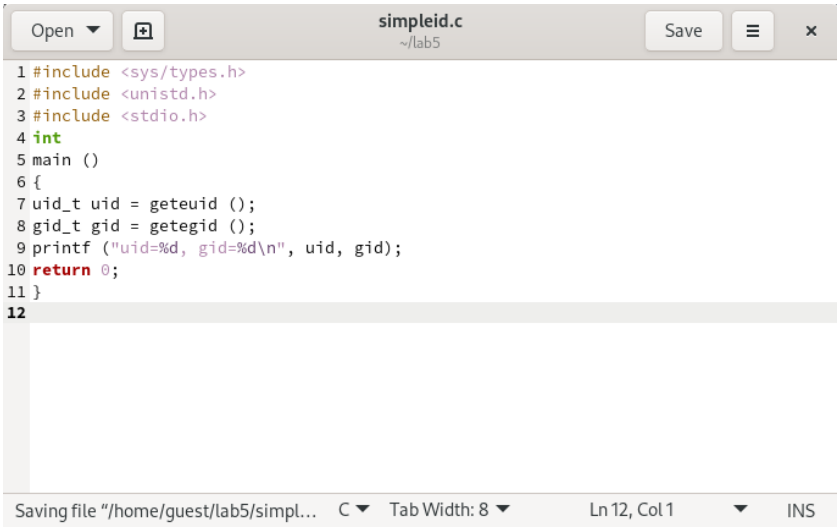

Рис. 2.2: Программа simpleid

3. Скомпилировали программу и убедились, что файл программы создан: `gcc simpleid.c -o simpleid`

4. Выполнили программу simpleid командой `./simpleid`

5. Выполнили системную программу id с помощью команды `id`. uid и gid совпадает в обеих программах

```
[guest@kreachna ~]$ mkdir lab5
[guest@kreachna ~]$ cd lab5
[guest@kreachna lab5]$ touch simpleid.c
[guest@kreachna lab5]$
[guest@kreachna lab5]$ gcc simpleid.c -o simpleid
[guest@kreachna lab5]$ ./simpleid
uid=1001, gid=1001
[guest@kreachna lab5]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r
:unconfined_t:s0-s0:c0.c1023
[guest@kreachna lab5]$
```

Рис. 2.3: Результат программы simpleid

6. Усложнили программу, добавив вывод действительных идентификаторов.

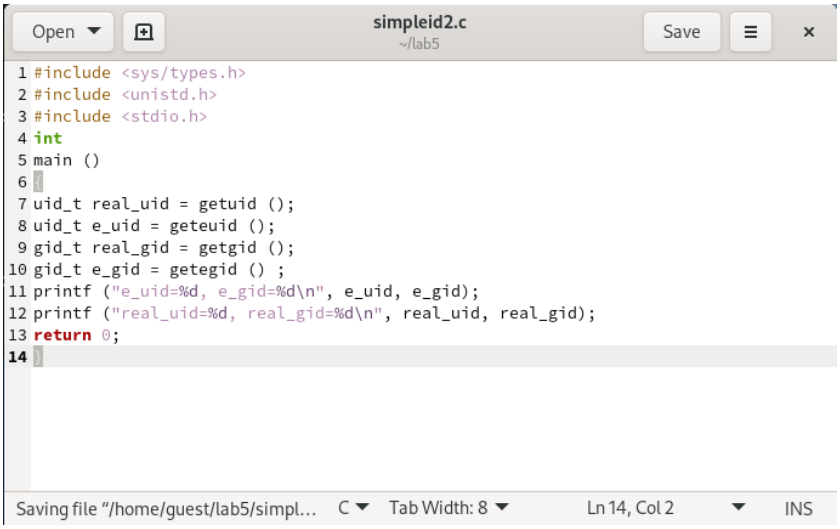

Рис. 2.4: Программа simpleid2

7. Скомпилировали и запустили simpleid2.c:

```
gcc simpleid2.c -o simpleid2
./simpleid2
```

8. От имени суперпользователя выполнили команды:

```
chown root:guest /home/guest/simpleid2
chmod u+s /home/guest/simpleid2
```

9. Использовали su для повышения прав до суперпользователя

10. Выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

```
ls -l simpleid2
```

11. Запустили simpleid2 и id:

```
./simpleid2
id
```

Результат выполнения программ теперь немного отличается

12. Проделали тоже самое относительно SetGID-бита.

Рис. 2.5: Результат программы simpleid2

13. Написали программу readfile.c

Рис. 2.6: Программа readfile

14. Откомпилировали её.

```
gcc readfile.c -o readfile
```

15. Сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.

```
chown root:guest /home/guest/readfile.c
chmod 700 /home/guest/readfile.c
```

16. Проверили, что пользователь guest не может прочитать файл readfile.c.
17. Сменили у программы readfile владельца и установили SetU'D-бит.
18. Проверили, может ли программа readfile прочитать файл readfile.c
19. Проверили, может ли программа readfile прочитать файл /etc/shadow

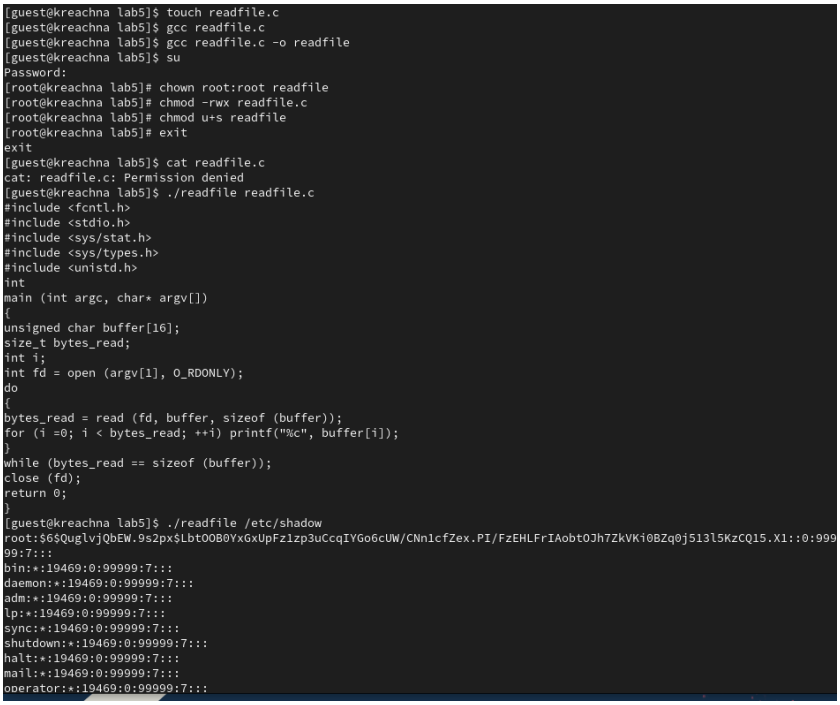

Рис. 2.7: Результат программы readfile

2.3 Исследование Sticky-бита

1. Выяснили, установлен ли атрибут Sticky на директории /tmp:

```
ls -l / | grep tmp
```

2. От имени пользователя guest создали файл file01.txt в директории /tmp со словом test:

```
echo "test" > /tmp/file01.txt
```

3. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные»:

```
ls -l /tmp/file01.txt
chmod o+rw /tmp/file01.txt
ls -l /tmp/file01.txt
```

Первоначально все группы имели право на чтение, а запись могли осуществлять все, кроме «остальных пользователей».

4. От пользователя (не являющегося владельцем) попробовали прочитать файл /file01.txt:

```
cat /file01.txt
```

5. От пользователя попробовали дозаписать в файл /file01.txt слово test3 командой:

```
echo "test2" >> /file01.txt
```

6. Проверили содержимое файла командой:

```
cat /file01.txt
```

В файле теперь записано:

```
Test
Test2
```

7. От пользователя попробовали записать в файл /tmp/file01.txt слово test4, стерев при этом всю имеющуюся в файле информацию командой. Для этого воспользовалась командой echo “test3” > /tmp/file01.txt

8. Проверили содержимое файла командой

```
cat /tmp/file01.txt
```

9. От пользователя попробовали удалить файл /tmp/file01.txt командой rm /tmp/file01.txt, однако получила отказ.

10. От суперпользователя командой выполнили команду, снимающую атрибут t (Sticky-бит) с директории /tmp:

```
chmod -t /tmp
```

Покинули режим суперпользователя командой exit.

11. От пользователя проверили, что атрибута t у директории /tmp нет:

```
ls -l / | grep tmp
```

12. Повторили предыдущие шаги. Получилось удалить файл

13. Удалось удалить файл от имени пользователя, не являющегося его владельцем.

14. Повысили свои права до суперпользователя и вернули атрибут t на директорию /tmp :

```
su
chmod +t /tmp
exit
```

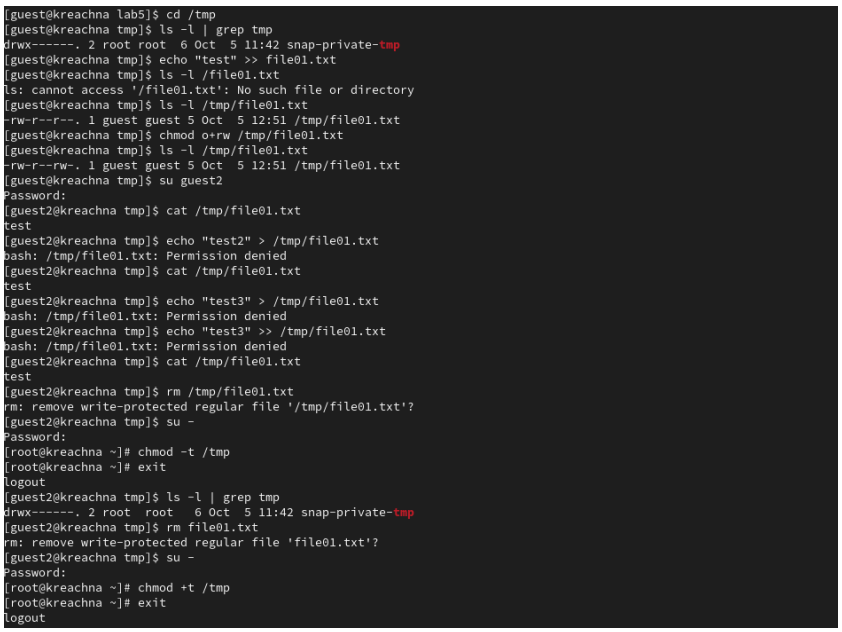

Рис. 2.8: Исследование Sticky-бита

3 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.

Список литературы

1. КОМАНДА CHATTR В LINUX
2. chattr