

Отчёт по лабораторной работе №5

Информационная безопасность

**Дискреционное разграничение прав в Linux. Исследование
влияния дополнительных атрибутов**

Выполнила: Абдуллина Ляйсан Раисовна,
НПИбд-01-21, 1032216538

Содержание

Цель работы	4
Теоретическое введение	5
Выполнение лабораторной работы	7
5.2.1. Подготовка лабораторного стенда	7
5.3.1 Создание программы	7
5.3.2. Исследование Sticky-бита	13
Вывод	17
Список литературы. Библиография	18

Список иллюстраций

1	(рис. 1. Установка gss)	7
2	(рис. 2. simpleid.c)	8
3	(рис. 3. 3-5 пункты задания лабораторной)	8
4	(рис. 4. simpleid2.c)	9
5	(рис. 5. 7 пункт задания лабораторной)	9
6	(рис. 6. 8-12 пункты задания лабораторной)	10
7	(рис. 7. readfile.c)	11
8	(рис. 8. chmod)	11
9	(рис. 9. 16-19 пункты Guest)	12
10	(рис. 10. 16-18 пункты суперпользователь)	12
11	(рис. 11. 19 пункт суперпользователь)	13
12	(рис. 12. 1-3 пункты)	14
13	(рис. 13. 4-12 пункты)	15
14	(рис. 14. Возвращение атрибута)	16

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

Теоретическое введение

1. Дополнительные атрибуты файлов Linux

В Linux существует три основных вида прав — право на чтение (read), запись (write) и выполнение (execute), а также три категории пользователей, к которым они могут применяться — владелец файла (user), группа владельца (group) и все остальные (others). Но, кроме прав чтения, выполнения и записи, есть еще три дополнительных атрибута. [1]

- **Sticky bit**

Используется в основном для каталогов, чтобы защитить в них файлы. В такой каталог может писать любой пользователь. Но, из такой директории пользователь может удалить только те файлы, владельцем которых он является. Примером может служить директория /tmp, в которой запись открыта для всех пользователей, но нежелательно удаление чужих файлов.

- **SUID (Set User ID)**

Атрибут исполняемого файла, позволяющий запустить его с правами владельца. В Linux приложение запускается с правами пользователя, запустившего указанное приложение. Это обеспечивает дополнительную безопасность т.к. процесс с правами пользователя не сможет получить доступ к важным системным файлам, которые принадлежат пользователю root.

- **SGID (Set Group ID)**

Аналогичен suid, но относится к группе. Если установить sgid для каталога, то все файлы созданные в нем, при запуске будут принимать идентификатор группы каталога, а не группы владельца, который создал файл в этом каталоге.

- **Обозначение атрибутов sticky, suid, sgid**

Специальные права используются довольно редко, поэтому при выводе программы `ls -l` символ, обозначающий указанные атрибуты, закрывает символ стандартных прав доступа.

Пример:

```
rwsrwsrwt
```

где первая s — это suid, вторая s — это sgid, а последняя t — это sticky bit

В приведенном примере не понятно, `rwt` — это `rw-` или `rwX`? Определить это просто. Если `t` маленькое, значит `x` установлен. Если `T` большое, значит `x` не установлен. То же самое правило распространяется и на `s`.

В числовом эквиваленте данные атрибуты определяются первым символом при четырехзначном обозначении (который часто опускается при назначении прав), например в правах `1777` — символ `1` обозначает sticky bit. Остальные атрибуты имеют следующие числовое соответствие:

1 — установлен sticky bit

2 — установлен sgid

4 — установлен suid

2. Компилятор GCC

`GCC` - это свободно доступный оптимизирующий компилятор для языков `C`, `C++`. Собственно программа `gcc` это некоторая надстройка над группой компиляторов, которая способна анализировать имена файлов, передаваемые ей в качестве аргументов, и определять, какие действия необходимо выполнить. Файлы с расширением `.cc` или `.C` рассматриваются, как файлы на языке `C++`, файлы с расширением `.c` как программы на языке `C`, а файлы с расширением `.o` считаются объектными. [2]


```

[guest@lrabdullina dir1]$ touch impleid.c
[guest@lrabdullina dir1]$ nano simpleid.c
[guest@lrabdullina dir1]$ ls simpleid.c
simpleid.c
[guest@lrabdullina dir1]$ cat simpleid.c
#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;
}
[guest@lrabdullina dir1]$ █

```

Рис. 2: (рис. 2. simpleid.c)

3. Скомпилируйте программу и убедитесь, что файл программы создан: `gcc simpleid.c -o simpleid`
4. Выполните программу simpleid: `./simpleid`
5. Выполните системную программу id: `id` и сравните полученный вами результат с данными предыдущего пункта задания.

```

}
[guest@lrabdullina dir1]$ gcc simpleid.c -o simpleid
[guest@lrabdullina dir1]$ ./simpleid
uid=1001, gid=1001
[guest@lrabdullina dir1]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined
_r:unconfined_t:s0-s0:c0.c1023
[guest@lrabdullina dir1]$ █

```

Рис. 3: (рис. 3. 3-5 пункты задания лабораторной)

6. Усложните программу, добавив вывод действительных идентификаторов.


```
[guest@lrabduullina dir1]$ touch simpleid2.c
[guest@lrabduullina dir1]$ nano simpleid2.c
[guest@lrabduullina dir1]$ ls simpleid2.c
simpleid2.c
[guest@lrabduullina dir1]$ cat simpleid2.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;
}
[guest@lrabduullina dir1]$
```

Рис. 4: (рис. 4. simpleid2.c)

7. Скомпилируйте и запустите simpleid2.c: gcc simpleid2.c -o simpleid2 ./simpleid2

```
[guest@lrabduullina dir1]$ gcc simpleid2.c -o simpleid2
[guest@lrabduullina dir1]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@lrabduullina dir1]$
```

Рис. 5: (рис. 5. 7 пункт задания лабораторной)

8. От имени суперпользователя выполните команды: chown root:guest /home/guest/simpleid2
chmod u+s /home/guest/simpleid2
9. Используйте sudo или повысьте временно свои права с помощью su. Поясните, что делают эти команды.

От имени суперпользователя выполнила команды “sudo chown root:guest /home/guest/simpleid2” и “sudo chmod u+s /home/guest/simpleid2”, затем выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой “sudo ls -l /home/guest/simpleid2” (рис. 3.9). Этими командами была произведена смена пользователя файла на root и установлен SetUID-бит.

10. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2: `ls -l simpleid2`
11. Запустите simpleid2 и id: `./simpleid2 id` Сравните результаты.
12. Прodelайте тоже самое относительно SetGID-бита.

```
[root@lrabduullina dir1]# chown root:guest /home/guest/dir1/simpleid2
[root@lrabduullina dir1]# chmod u+s /home/guest/dir1/
dir2/      impleid.c      simpleid2      simpleid.c
file1      simpleid      simpleid2.c
[root@lrabduullina dir1]# chmod u+s /home/guest/dir1/simpleid2
[root@lrabduullina dir1]# s -l simpleid2
bash: s: команда не найдена...
[root@lrabduullina dir1]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 17720 окт  4 20:05 simpleid2
[root@lrabduullina dir1]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@lrabduullina dir1]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfine
d_t:s0-s0:c0.c1023
[root@lrabduullina dir1]#
```

Рис. 6: (рис. 6. 8-12 пункты задания лабораторной)

13. Создайте программу readfile.c
14. Откомпилируйте её. `gcc readfile.c -o readfile`

```

[root@lrabdullina dir1]# touch readfile.c
[root@lrabdullina dir1]# nano readfile.c
[root@lrabdullina dir1]# ls readfile.c
readfile.c
[root@lrabdullina dir1]# cat 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;
}
[root@lrabdullina dir1]# gcc readfile.c -o readfile
[root@lrabdullina dir1]#

```

Рис. 7: (рис. 7. readfile.c)

15. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.

```

[root@lrabdullina dir1]# gcc readfile.c -o readfile
[root@lrabdullina dir1]# chown root:guest readfile
[root@lrabdullina dir1]# chown 700 readfile
[root@lrabdullina dir1]# chown -r readfile.c
chown: неверный ключ - «r»
По команде «chown --help» можно получить дополнительную информацию.
[root@lrabdullina dir1]# chmod -r readfile.c
[root@lrabdullina dir1]# chmod u+s readfile
[root@lrabdullina dir1]#

```

Рис. 8: (рис. 8. chmod)

16. Проверьте, что пользователь guest не может прочитать файл readfile.c.
17. Смените у программы readfile владельца и установите SetU'D-бит.

18. Проверьте, может ли программа readfile прочитать файл readfile.c?
19. Проверьте, может ли программа readfile прочитать файл /etc/shadow? Отрадите полученный результат и ваши объяснения в отчёте.

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

Рис. 9: (рис. 9. 16-19 пункты Guest)

От имени суперпользователя все команды удастся выполнить.

```
[guest@lrabdullina dir1]$ su
Пароль:
[root@lrabdullina dir1]# cat 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. 10: (рис. 10. 16-18 пункты суперпользователь)

```

dSYSTEMD_EXEC_PID=2191XAUTHORITY=/root/.xauthH5mgC0GDM_LANG=ru_RU.UTF-8HOME=/rootU
ERNAME=guestLANG=ru_RU.UTF-8LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;
5:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=01;37;41:su=37;41:sg=30;43:ca=30
41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=0
;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz
01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01
31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;
1:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01
31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=
1;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=0
;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.
bm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.
tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;3
:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=
1;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv
01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.fl
=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf
01;35:*.ogv=01;35:*.ogx=01;35:*.aac=01;36:*.au=01;36:*.flac=01;36:*.m4a=01;36:*.mi
=01;36:*.midi=01;36:*.mka=01;36:*.mp3=01;36:*.mpc=01;36:*.ogg=01;36:*.ra=01;36:*.w
v=01;36:*.oga=01;36:*.opus=01;36:*.spx=01;36:*.xspf=01;36:XDG_CURRENT_DESKTOP=GNOM
VTE_VERSION=6402WAYLAND_DISPLAY=wayland-0GNOME_TERMINAL_SCREEN=/org/gnome/Terminal
screen/d8dcfa1c_698c_448e_9502_0fafbf6e3898GNOME_SETUP_DISPLAY=:1XDG_SESSION_CLASS
userTERM=xterm-256colorLESSOPEN=||/usr/bin/lesspipe.sh %sUSER=guestGNOME_TERMINAL_
ERVICE=:1.109DISPLAY=:0SHLVL=2QT_IM_MODULE=ibusXDG_RUNTIME_DIR=/run/user/1001DEBUG
NFOD_URLS=https://debuginfod.centos.org/ which_declare=declare -fxDG_DATA_DIRS=/ro
ot/.local/share/flatpak/exports/share:/home/guest/.local/share/flatpak/exports/shar
e:/var/lib/flatpak/exports/share:/usr/local/share:/usr/share/PATH=/root/.local/bin
/root/bin:/home/guest/.local/bin:/home/guest/bin:/usr/local/bin:/usr/local/sbin:/u
r/bin:/usr/sbinGDMSESSION=gnomeDBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1001/b
sMAIL=/var/spool/mail/guestOLDPWD=/home/guestBASH_FUNC_which%%=( { ( alias;
eval ${which_declare} ) | /usr/bin/which --tty-only --read-alias --read-functions
--show-tilde --show-dot $@
Ошибка сегментирования (стек памяти сброшен на диск)

```

Рис. 11: (рис. 11. 19 пункт суперпользователь)

5.3.2. Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду `ls -l / | grep tmp`
2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test: `echo "test" > /tmp/file01.txt`
3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»: `ls -l /tmp/file01.txt chmod o+rw /tmp/file01.txt ls -l /tmp/file01.txt`

```

[guest@lrabdu1lina dir1]$ ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 окт  4 20:16 tmp
[guest@lrabdu1lina dir1]$ echo "test" > /tmp/file01.txt
[guest@lrabdu1lina dir1]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 окт  4 20:16 /tmp/file01.txt
[guest@lrabdu1lina dir1]$ chmod o+rw /tmp/file01.txt
[guest@lrabdu1lina dir1]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 окт  4 20:16 /tmp/file01.txt
[guest@lrabdu1lina dir1]$

```

Рис. 12: (рис. 12. 1-3 пункты)

4. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt: `cat /tmp/file01.txt`
5. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой `echo "test2" > /tmp/file01.txt`

Удалось ли вам выполнить операцию? Нет.

6. Проверьте содержимое файла командой `cat /tmp/file01.txt`
7. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt`

Удалось ли вам выполнить операцию? Нет.

8. Проверьте содержимое файла командой `cat /tmp/file01.txt`
9. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt`

Удалось ли вам удалить файл? Нет.

10. Повысьте свои права до суперпользователя следующей командой `su` и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: `chmod -t /tmp`

11. Покиньте режим суперпользователя командой `exit`
12. От пользователя `guest2` проверьте, что атрибута `t` у директории `/tmp` нет: `ls -l / | grep tmp`

```
guest@lrabdullina dir1]$ ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 окт  4 20:16 tmp
guest@lrabdullina dir1]$ echo "test" > /tmp/file01.txt
guest@lrabdullina dir1]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 окт  4 20:16 /tmp/file01.txt
guest@lrabdullina dir1]$ chmod o+rw /tmp/file01.txt
guest@lrabdullina dir1]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 окт  4 20:16 /tmp/file01.txt
guest@lrabdullina dir1]$ cat /tmp/file01.txt
test
guest@lrabdullina dir1]$ su guest2
su: user guest2 does not exist or the user entry does not contain all the required
fields
guest@lrabdullina dir1]$ cat /tmp/file01.txt
test
guest@lrabdullina dir1]$ echo "test2" > /tmp/file01.txt
guest@lrabdullina dir1]$ cat /tmp/file01.txt
test2
guest@lrabdullina dir1]$ echo "test" > /tmp/file01.txt
guest@lrabdullina dir1]$ cat /tmp/file01.txt
test
guest@lrabdullina dir1]$ echo "test3" > /tmp/file01.txt
guest@lrabdullina dir1]$ echo "test" > /tmp/file01.txt
guest@lrabdullina dir1]$ cat /tmp/file01.txt
test
guest@lrabdullina dir1]$ su -
пароль:
root@lrabdullina ~]# chmod -t /tmp
root@lrabdullina ~]# exit
выход
guest@lrabdullina dir1]$ ls -l / | grep tmp
drwxrwxrwx. 19 root root 4096 окт  4 20:21 tmp
guest@lrabdullina dir1]$
```

Рис. 13: (рис. 13. 4-12 пункты)

13. Повторите предыдущие шаги. Какие наблюдаются изменения?

При повторении всё получилось.

14. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Удалось.
15. Повысьте свои права до суперпользователя и верните атрибут `t` на директорию `/tmp`:
`su chmod +t /tmp exit`

```
[guest@lrabdullina dir1]$ su -  
Пароль:  
[root@lrabdullina ~]# chmod +t /tmp  
[root@lrabdullina ~]# exit  
выход  
[guest@lrabdullina dir1]$
```

Рис. 14: (рис. 14. Возвращение атрибута)

Вывод

Были изучены механизмы изменения идентификаторов и применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами. Были рассмотрены работа механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов

Список литературы. Библиография

[0] Методические материалы курса

[1] Дополнительные атрибуты: <https://tokmakov.msk.ru/blog/item/141>

[2] Компилятор GSS: <http://parallel.imm.uran.ru/freesoft/make/instrum.html>