

Отчёт по лабораторной работе №5

**Дискреционное Разграничение Прав в Linux. Исследование
Влияния Дополнительных Атрибутов**

Вакутайпа Милдред

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	13
	Список литературы	14

Список иллюстраций

2.1	Проверка рабочее пространство	6
2.2	Создание simpleid.c	6
2.3	simpleid2.c	8
2.4	Вывод simpleid2.c	8
2.5	Изменение права доступа на simpleid2.c	8
2.6	сравнение simpleid2 на id	8
2.7	Программу readfile.c	9
2.8	Смена владельца файла и прав доступа	9
2.9	Попытка прочесть файл	10
2.10	Попытка 3 прочесть файл	10
2.11	Чтение файла от суперпользователя	10
2.12	Создание файла	10
2.13	изменение права доступа	11
2.14	Проверка файл на чтение, запись и удаление	11
2.15	Снятие атрибута Sticky	11
2.16	Повтор предыдущих действий	12
2.17	Изменение атрибутов	12

Список таблиц

1 Цель работы

Изучение механизмов изменения идентификаторов приложения 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

До начала работы проверила, что имеется средства разработки:

```
[mwakutaipa@mwakutaipa ~]$ 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-now --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share
/man --infodir=/usr/share/info --with-bugurl=https://bugs.rockylinux.org/ --enab
le-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-in
itfini-array --without-isl --enable-multilib --with-linker-hash-style=gnu --enab
le-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-functi
on --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-s
erialization=1
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.5.0 20240719 (Red Hat 11.5.0-5) (GCC)
[mwakutaipa@mwakutaipa ~]$ sudo setenforce 0
```

Рис. 2.1: Проверка рабочее пространство

Вошла в систему от имени пользователя guest и создала программу simpleid.c:

```
[mwakutaipa@mwakutaipa ~]$ su guest
Password:
[guest@mwakutaipa mwakutaipa]$ cd
[guest@mwakutaipa ~]$ touch simpleid.c
[guest@mwakutaipa ~]$ nano simpleid.c
```

Рис. 2.2: Создание simpleid.c

“ С++ листинг 1

```
#include <sys/types.h> #include <unistd.h> #include <stdio.h>
```

```
int main(){ uid_t uid = getuid(); gid_t gid = getgid(); printf("uid=%d,gid=%d\n", uid,
gid); return 0; }
```

![код программы 1](image/3.jpg){#fig:003 width=70%}

Скомпилировала программу и запускаю ее. Она выводит идентификатор пользователя и групп

![Вывод simpleid.c](image/4.jpg){#fig:004 width=70%}

Создала файл simpleid2.c добавив вывод действительных идентификаторов.

``` C++ листинг 2

```
#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@mwakutaipa ~]$ touch simpleid2.c
[guest@mwakutaipa ~]$ nano simpleid2.c
```

Рис. 2.3: simpleid2.c

Скомпилировала программу и запускаю ее.

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

Рис. 2.4: Вывод simpleid2.c

С помощью `chown` изменяю владельца файла на суперпользователя, с помощью `chmod` изменяю права доступа

```
[guest@mwakutaipa ~]$ sudo chown root:guest /home/guest/simpleid2
[sudo] password for guest:
[guest@mwakutaipa ~]$ sudo chmod u+s /home/guest/simpleid2
[guest@mwakutaipa ~]$ ls -l
total 68
-rwxr-xr-x. 1 guest guest 17600 Apr 17 17:30 a.out
drwxr-xr-x. 2 guest guest 6 Mar 15 18:27 check
drwxr-xr-x. 2 guest guest 19 Apr 1 21:07 dir1
-rwxr-xr-x. 1 guest guest 17600 Apr 17 17:31 simpleid
-rwsr-xr-x. 1 root guest 17704 Apr 17 17:41 simpleid2
```

Рис. 2.5: Изменение права доступа на simpleid2.c

Сравнение вывода программы и команды `id`, наша программа вывела только ограниченное количество информации

```
[guest@mwakutaipa ~]$ sudo /home/guest/simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[guest@mwakutaipa ~]$ sudo id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
```

Рис. 2.6: сравнение simpleid2 на id

Создала еще одну программу `readfile.c`

C++ листинг 3 `#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; }

```

```

GNU nano 5.6.1 readfile.c Modified
#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. 2.7: Программу readfile.c

Снова от имени суперпользователя изменила владельца файла readfile. Далее изменила права доступа так, чтобы пользователь guest не смог прочесть содержимое файла

```

[guest@mwakutaipa ~]$ sudo chown root:guest /home/guest/readfile.c
[guest@mwakutaipa ~]$ sudo chmod u+s /home/guest/readfile.c
[guest@mwakutaipa ~]$ sudo chmod 700 /home/guest/readfile.c
[guest@mwakutaipa ~]$ sudo chmod -r /home/guest/readfile.c
[guest@mwakutaipa ~]$ sudo chmod u+s /home/guest/readfile.c
[guest@mwakutaipa ~]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@mwakutaipa ~]$

```

Рис. 2.8: Смена владельца файла и прав доступа

Проверка прочесть файл от имени пользователя guest. Прочесть файл не удастся. Попытка прочесть тот же файл с помощью программы readfile, в ответ получаем “отказано в доступе”



Рис. 2.9: Попытка прочесть файл

Попытка прочесть файл shadow с помощью программы, все еще получаем отказ в доступе

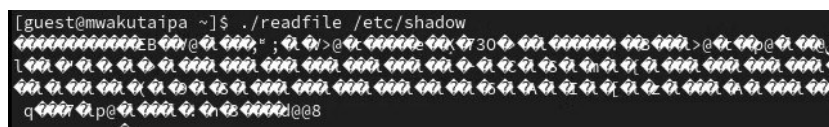


Рис. 2.10: Попытка 3 прочесть файл

Попробовала прочесть эти же файлы от имени суперпользователя и чтение файлов было успешно



Рис. 2.11: Чтение файла от суперпользователя

От имени пользователя guest создаю файл с текстом.

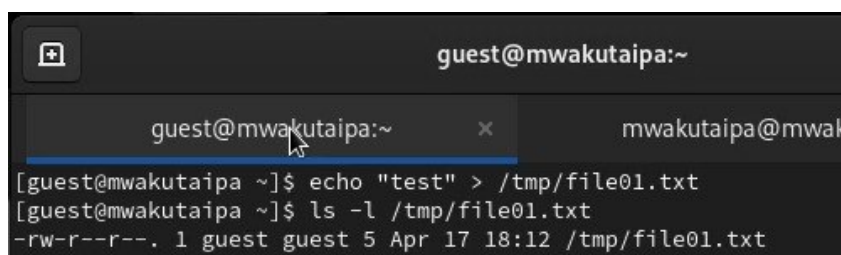


Рис. 2.12: Создание файла

Добавляю права на чтение и запись для других пользователей

```
[guest@mwakutaipa ~]$ chmod o+rw /tmp/file01.txt
[guest@mwakutaipa ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Apr 17 18:12 /tmp/file01.txt
[guest@mwakutaipa ~]$
```

Рис. 2.13: изменение права доступа

Вхожу в систему от имени пользователя guest2, от его имени могу прочитать файл file01.txt, но перезаписать информацию в нем не могу. Также невозможно добавить в файл file01.txt новую информацию от имени пользователя guest2. Когда попробовала удалить файл, снова получила отказ.

```
[guest2@mwakutaipa mwakutaipa]$ cat /tmp/file01.txt
test
[guest2@mwakutaipa mwakutaipa]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@mwakutaipa mwakutaipa]$ cat /tmp/file01.txt
test
[guest2@mwakutaipa mwakutaipa]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@mwakutaipa mwakutaipa]$ cat /tmp/file01.txt
test
[guest2@mwakutaipa mwakutaipa]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'?
[guest2@mwakutaipa mwakutaipa]$ ls
ls: cannot open directory '.': Permission denied
[guest2@mwakutaipa mwakutaipa]$
```

Рис. 2.14: Проверка файл на чтение, запись и удаление

От имени суперпользователя сняла с директории атрибут Sticky.

```
[guest2@mwakutaipa mwakutaipa]$ su -
Password:
[root@mwakutaipa ~]# chmod -t /tmp
[root@mwakutaipa ~]# exit
logout
```

Рис. 2.15: Снятие атрибута Sticky

Проверила, что атрибут действительно снят и выполнен повтор предыдущих действий. По результатам без Sticky-бита запись в файл и дозапись в файл осталась невозможной

```
[guest2@mwakutaipa mwakutaipa]$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 Apr 17 18:33 tmp
[guest2@mwakutaipa mwakutaipa]$ cat /tmp/file01.txt
test
[guest2@mwakutaipa mwakutaipa]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@mwakutaipa mwakutaipa]$ cat /tmp/file01.txt
test
[guest2@mwakutaipa mwakutaipa]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@mwakutaipa mwakutaipa]$ cat /tmp/file01.txt
test
[guest2@mwakutaipa mwakutaipa]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'?
[guest2@mwakutaipa mwakutaipa]$ ls -l / | grep tmp
drwxrwxrwx. 18 root root 4096 Apr 17 18:33 tmp
```

Рис. 2.16: Повтор предыдущих действий

Возвращение директории tmp атрибута t от имени суперпользователя

```
[guest2@mwakutaipa mwakutaipa]$ su -
Password:
[root@mwakutaipa ~]# chmod +t /tmp
[root@mwakutaipa ~]# exit
logout
```

Рис. 2.17: Изменение атрибутов

## 3 Выводы

Изучила механизм изменения идентификаторов, применила SetUID- и Sticky-биты. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

## **Список литературы**