

Отчет по лабораторной работе №5

Дискреционное разграничение прав в Linux. Исследование влияния
дополнительных атрибутов

Поленикова Анна Алексеевна

Содержание

Цель работы	4
Выполнение лабораторной работы	5
Подготовка лабораторного стенда	5
Создание программы	5
Исследование Sticky-бита	10
Вывод	13

Список иллюстраций

0.1	Подготовка	5
0.2	Код simpleid.c	6
0.3	Компиляция и выполнение программы simpleid	6
0.4	Код simpleid2.c	7
0.5	Компиляция и выполнение программы simpleid2	7
0.6	Изменение владельца и атрибутов simpleid2	7
0.7	Запуск simpleid2 и id	8
0.8	Повторение операций для SetGID-бита	8
0.9	Код readfile.c	8
0.10	Компиляция. Смена владельца и изменение прав файла readfile.c . . .	9
0.11	Смена владельца программы readfile и установка SetUID-бита. Чтение файла readfile.c	9
0.12	Чтение файла /etc/shadow	9
0.13	Проверка Sticky атрибута. Создание файла file1. Изменение атрибутов для категории пользователей «все остальные»	10
0.14	Операции с файлом file01.txt	11
0.15	Снятие t атрибута с директории /tmp	11
0.16	Операции с файлом file01.txt	12
0.17	Возвращение t атрибута директории /tmp	12

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.

Выполнение лабораторной работы

Подготовка лабораторного стенда

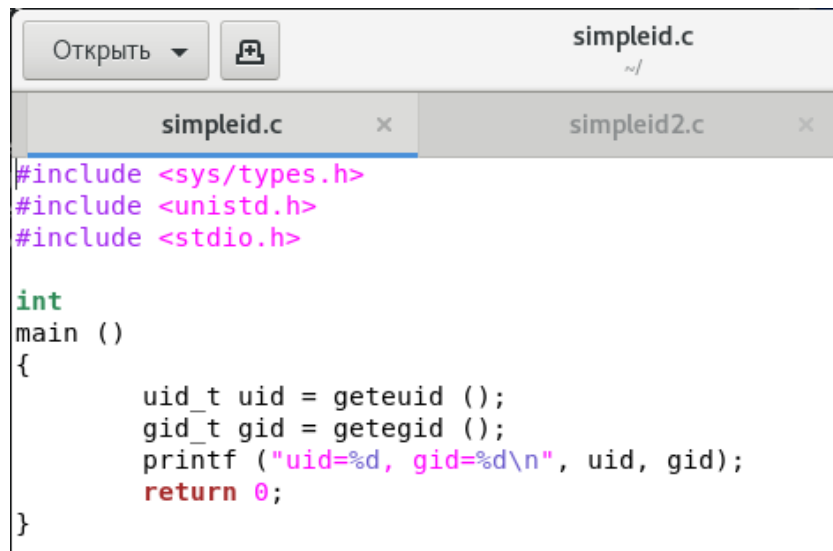
1. Убедилась, что в системе установлен компилятор gcc, введя команду `gcc -v`.
Также проверила отключение систему запретов до очередной перезагрузки системы командой `getenforce`, которая вывела `Permissive`.

```
[guest@aapolenikova ~]$ gcc -v
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/8/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ./configure --enable-bootstrap --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-shared --enable-threads=posix --enable-checking=release --enable-multilib --with-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-gcc-major-version-only --with-linker-hash-style=gnu --enable-plugin --enable-initfini-array --with-isl --disable-libmpx --enable-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-function --enable-cet --with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux
Модель многопоточности: posix
gcc версия 8.5.0 20210514 (Red Hat 8.5.0-3) (GCC)
[guest@aapolenikova ~]$ getenforce
Permissive
```

Рис. 0.1: Подготовка

Создание программы

1. Вошла в систему от имени пользователя `guest` и создала программу `simpleid.c` со следующим кодом:




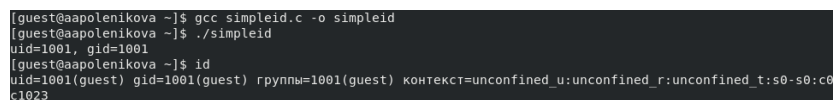
```
Открыть ▾  simpleid.c ~/  
simpleid.c × simpleid2.c ×  
#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
  
int  
main ()  
{  
    uid_t uid = geteuid ();  
    gid_t gid = getegid ();  
    printf ("uid=%d, gid=%d\n", uid, gid);  
    return 0;  
}
```

Рис. 0.2: Код simpleid.c

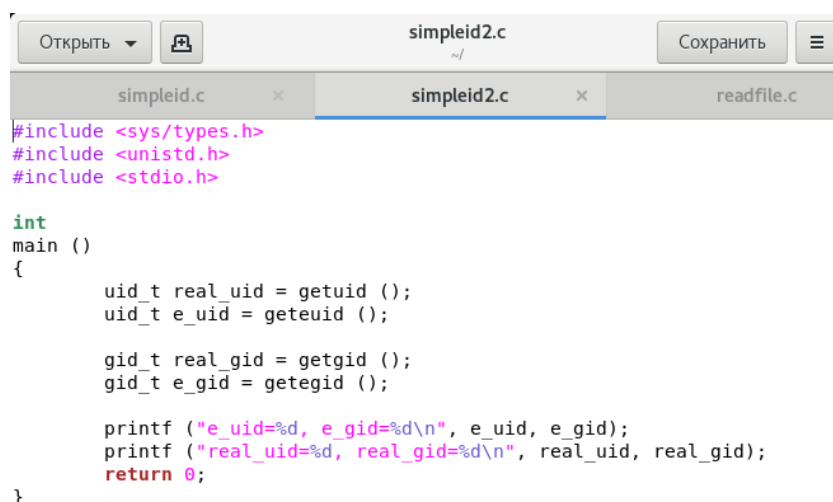
2. Скомпилировала программу с помощью команды `gcc simpleid.c -o simpleid` и убедилась, что файл программы создан. Выполнила программу `simpleid`, а также системную программу `id`. Результат выполнения двух последних программ одинаков.



```
[guest@aapolenikova ~]$ gcc simpleid.c -o simpleid  
[guest@aapolenikova ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@aapolenikova ~]$ id  
uid=1001(guest) gid=1001(guest) rpnпы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 0.3: Компиляция и выполнение программы simpleid

3. Усложнила программу, добавив вывод действительных идентификаторов и назвала получившуюся программу `simpleid2.c`.



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

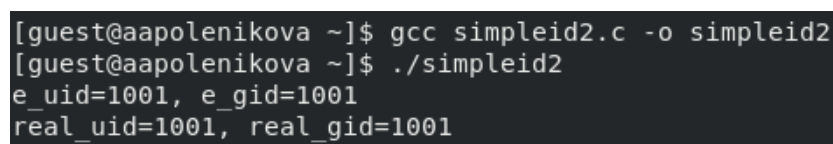
int
main ()
{
    uid_t real_uid = getuid ();
    uid_t e_uid = geteuid ();

    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;
}
```

Рис. 0.4: Код simpleid2.c

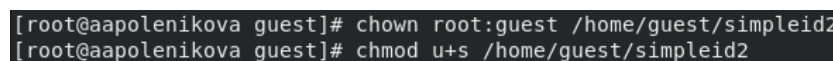
4. Скомпилировала и запустила simpleid2.c.



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

Рис. 0.5: Компиляция и выполнение программы simpleid2

5. От имени суперпользователя выполните команды `chown root:guest /home/guest/simpleid2` и `chmod u+s /home/guest/simpleid2`, повысив права пользователя с помощью команды `su` и изменив владельца и атрибуты simpleid2.



```
[root@aapolenikova guest]# chown root:guest /home/guest/simpleid2
[root@aapolenikova guest]# chmod u+s /home/guest/simpleid2
```

Рис. 0.6: Изменение владельца и атрибутов simpleid2

6. Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой `ls -l simpleid2`, а также запустила simpleid2 и

id. Результат выполнения программ отличается, поскольку программа simpleid2 выводит uid и gid владельца, а команда id - uid и gid текущего пользователя.

```
[guest@aapolenikova ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 17696 ноя 13 17:35 simpleid2
[guest@aapolenikova ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aapolenikova ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

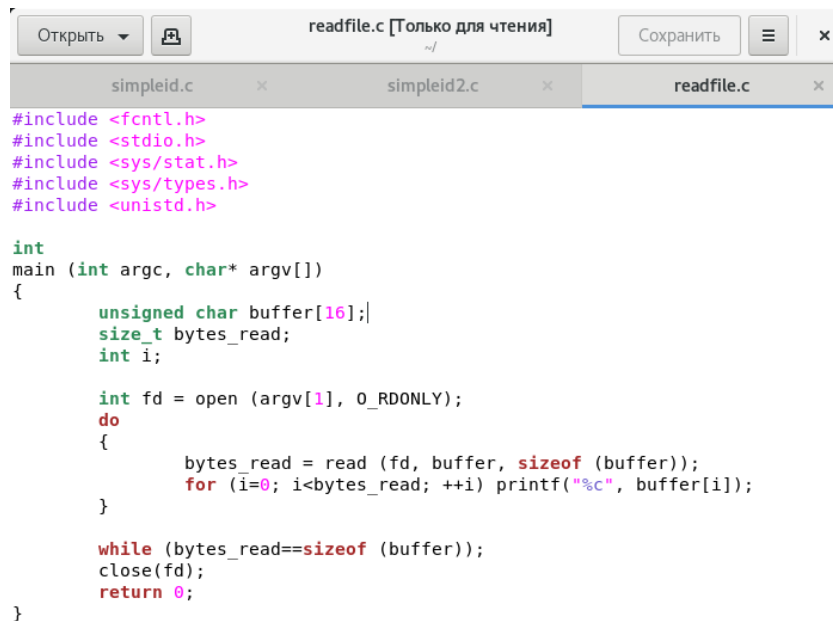
Рис. 0.7: Запуск simpleid2 и id

7. Проделала то же самое относительно SetGID-бита.

```
[root@aapolenikova guest]# chmod g+s /home/guest/simpleid2
[root@aapolenikova guest]# exit
exit
[guest@aapolenikova ~]$ ls -l simpleid2
-rwsrwsr-x. 1 root guest 17696 ноя 13 17:35 simpleid2
[guest@aapolenikova ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aapolenikova ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 0.8: Повторение операций для SetGID-бита

8. Создала 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;
}
```

Рис. 0.9: Код readfile.c

9. Откомпилируйте программу с помощью команды `gcc readfile.c -o readfile`. После от имени администратора сменила владельца у файла `readfile.c` и изменила права так, чтобы только суперпользователь (`root`) мог прочитать его, а `guest` не мог. Проверила, что пользователь `guest` не может прочитать файл `readfile.c`.

```
[guest@aapolenikova ~]$ gcc readfile.c -o readfile
[guest@aapolenikova ~]$ su
Пароль:
[root@aapolenikova guest]# chown root:guest /home/guest/readfile.c
[root@aapolenikova guest]# chmod ug-r /home/guest/readfile.c
[root@aapolenikova guest]# exit
exit
[guest@aapolenikova ~]$ ls -l readfile.c
--w----r--. 1 root guest 414 ноя 13 18:17 readfile.c
[guest@aapolenikova ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
```

Рис. 0.10: Компиляция. Смена владельца и изменение прав файла readfile.c

10. Сменила у программы readfile владельца и установила SetUID-бит. Выяснила, что программа readfile не может прочитать файл readfile.c.

```
[root@aapolenikova guest]# chmod u+s /home/guest/readfile
[root@aapolenikova guest]# chmod u-s /home/guest/readfile.c
[root@aapolenikova guest]# chown guest:root /home/guest/readfile
[root@aapolenikova guest]# chown root:guest /home/guest/readfile.c
[root@aapolenikova guest]# exit
exit
[guest@aapolenikova ~]$ ./readfile readfile.c
```

Рис. 0.11: Смена владельца программы readfile и установка SetUID-бита. Чтение файла readfile.c

11. Также выяснила, что программа readfile не может прочитать файл /etc/shadow.

[illegible]

Рис. 0.12: Чтение файла /etc/shadow

Исследование Sticky-бита

1. Выяснила, установлен ли атрибут Sticky на директории /tmp, для чего выполнила команду `ls -l / | grep tmp`. Атрибут установлен. От имени пользователя guest создала файл file01.txt в директории /tmp со словом test с помощью команды `echo "test" > /tmp/file01.txt`. После просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные» с помощью команд `ls -l /tmp/file01.txt`, `chmod o+rw /tmp/file01.txt` и `ls -l /tmp/file01.txt`.

```
[guest@aapolenikova ~]$ ls -l / | grep tmp
drwxrwxrwt. 13 root root 4096 ноя 13 18:41 tmp
[guest@aapolenikova ~]$ echo "test" > /tmp/file01.txt
[guest@aapolenikova ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 13 18:57 /tmp/file01.txt
[guest@aapolenikova ~]$ chmod o+rw /tmp/file01.txt
[guest@aapolenikova ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 13 18:57 /tmp/file01.txt
```

Рис. 0.13: Проверка Sticky атрибута. Создание файла file1. Изменение атрибутов для категории пользователей «все остальные»

2. От пользователя guest2 (не являющегося владельцем) попробовала прочитать файл /tmp/file01.txt с помощью команды `cat /tmp/file01.txt`. После попробовала дозаписать в файл /tmp/file01.txt слово test2 командой `echo "test2" > /tmp/file01.txt`. Проверила содержимое файла командой `cat /tmp/file01.txt`. Далее попробовала записать в файл /tmp/file01.txt слово test3, стерева при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt`. Снова проверила содержимое файла командой `cat /tmp/file01.txt`. Попробовала удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt`. Получилось выполнить все команды на запись и чтение, но не команду удаления файла.

```
[guest2@aapolenikova guest]$ cat /tmp/file01.txt
test
[guest2@aapolenikova guest]$ echo "test" > /tmp/file01.txt
[guest2@aapolenikova guest]$ cat /tmp/file01.txt
test
[guest2@aapolenikova guest]$ echo "test2" > /tmp/file01.txt
[guest2@aapolenikova guest]$ cat /tmp/file01.txt
test2
[guest2@aapolenikova guest]$ echo "test3" > /tmp/file01.txt
[guest2@aapolenikova guest]$ cat /tmp/file01.txt
test3
[guest2@aapolenikova guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
```

Рис. 0.14: Операции с файлом file01.txt

3. Повысила свои права до суперпользователя командой `su -` и выполнила команду `chmod -t /tmp`, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`. Покинула режим суперпользователя командой `exit`.

```
[guest2@aapolenikova guest]$ su -
Пароль:
[root@aapolenikova ~]# chmod -t /tmp
[root@aapolenikova ~]# exit
```

Рис. 0.15: Снятие `t` атрибута с директории `/tmp`

4. От пользователя `guest2` проверила командой `ls -l / | grep tmp`, что атрибута `t` у директории `/tmp` нет. Повторила предыдущие шаги, причем в этот раз удалось удалить файл от имени пользователя, не являющегося владельцем файла `file01.txt`.

```
[guest2@aapolenikova guest]$ ls -l / | grep tmp
drwxrwxrwx. 13 root root 4096 ноя 13 19:03 tmp
[guest2@aapolenikova guest]$ cat /tmp/file01.txt
test3
[guest2@aapolenikova guest]$ echo "test" > /tmp/file01.txt
[guest2@aapolenikova guest]$ cat /tmp/file01.txt
test
[guest2@aapolenikova guest]$ echo "test2" > /tmp/file01.txt
[guest2@aapolenikova guest]$ cat /tmp/file01.txt
test2
[guest2@aapolenikova guest]$ rm /tmp/file01.txt
```

Рис. 0.16: Операции с файлом file01.txt

5. Повысила свои права до суперпользователя и вернула атрибут `t` на директорию `/tmp`.

```
[guest2@aapolenikova guest]$ su -
Пароль:
[root@aapolenikova ~]# chmod +t /tmp
[root@aapolenikova ~]# exit
выход
```

Рис. 0.17: Возвращение `t` атрибута директории `/tmp`

Вывод

В ходе выполнения лабораторной работы были изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов, получены практические навыки работы в консоли с дополнительными атрибутами, а также рассмотрены работы механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.