

Отчёт по лабораторной работе № 5

Информационная безопасность

Арбатова Варвара Петровна

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Выводы	16
	Список литературы	17

Список таблиц

Список иллюстраций

3.1	Проверка	8
3.2	Создание файла	8
3.3	Текст файла	9
3.4	Работа с файлом	9
3.5	Текст файла	10
3.6	Работа с файлом	10
3.7	Изменение прав доступа	10
3.8	Сравнение выводов	11
3.9	Текст файла	11
3.10	Подготовка файла	11
3.11	Изменение прав	12
3.12	Попытка прочесть файл	12
3.13	Попытка прочесть файл	12
3.14	Попытка прочесть файл	12
3.15	Чтение файлов	13
3.16	Атрибут установлен	13
3.17	Изменение прав	13
3.18	Эксперименты	14
3.19	Снятие атрибута	14
3.20	Проверка	14
3.21	Эксперименты 2	15

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 Теоретическое введение

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. [@u]

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 `gwx`? 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` 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 `[@gcc]`.

3 Выполнение лабораторной работы

Для лабораторной работы необходимо проверить, установлен ли компилятор gcc, команда `gcc -v` позволяет это сделать. Также осуществляется отключение системы запретов с помощью `setenforce 0`

```
[vparbatova@vparbatova guest]$ whereis gcc
gcc: /etc/gcc
[vparbatova@vparbatova guest]$ whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/libexec/gcc /usr/share/man/man1/gcc.1.gz /usr/share/info/gcc.info.gz
[vparbatova@vparbatova guest]$ whereis g++
g++: /usr/bin/g++ /usr/share/man/man1/g++.1.gz
[vparbatova@vparbatova guest]$ gcc -v
Используются внутренние спецификации.
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
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ./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/ --enable-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-initfini-array --without-lto --enable-multilib --with-linker-hash-style=gnu --enable-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-function --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-lt0 --enable-link-serialization=1
Модель многопоточности: posix
Supported LTO compression algorithms: zlib zstd
gcc версия 11.5.0 20240719 (Red Hat 11.5.0-2) (GCC)
[vparbatova@vparbatova guest]$ sudo setenforce 0
[sudo] пароль для vparbatova:
Попройте ещё раз.
[sudo] пароля для vparbatova:
sudo: setenforce: command not found
[vparbatova@vparbatova guest]$ sudo setenforce 0
[vparbatova@vparbatova guest]$ getenforce
Permissive
[vparbatova@vparbatova guest]$
```

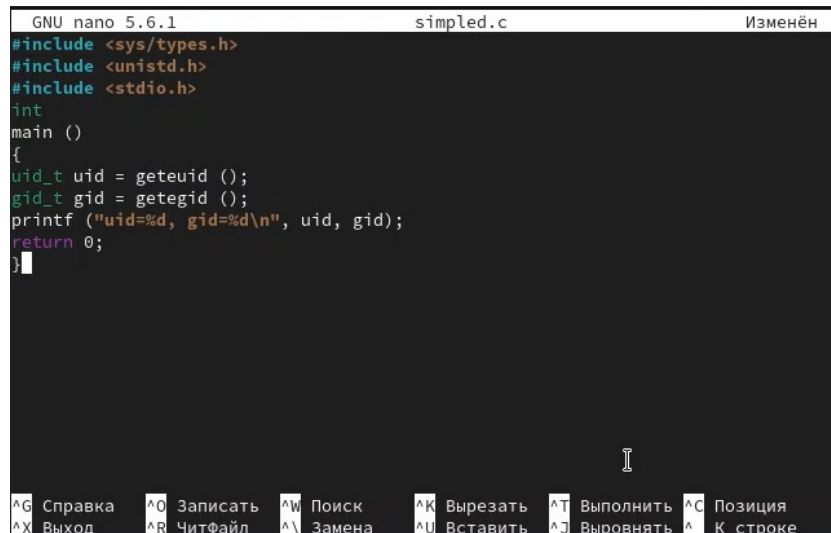
Рис. 3.1: Проверка

Создаю файл и открываю его в редакторе nano

```
[guest@vparbatova ~]$ touch simplified.c
[guest@vparbatova ~]$ nano simplified.c
```

Рис. 3.2: Создание файла

Текст файла

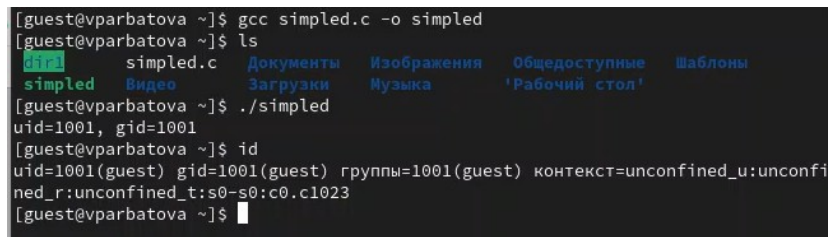


```
GNU nano 5.6.1 simplified.c Изменён
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main ()
{
  uid_t uid = geteuid ();
  gid_t gid = getegid ();
  printf ("uid=%d, gid=%d\n", uid, gid);
  return 0;
}
```

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выровнять ^_ К строке

Рис. 3.3: Текст файла

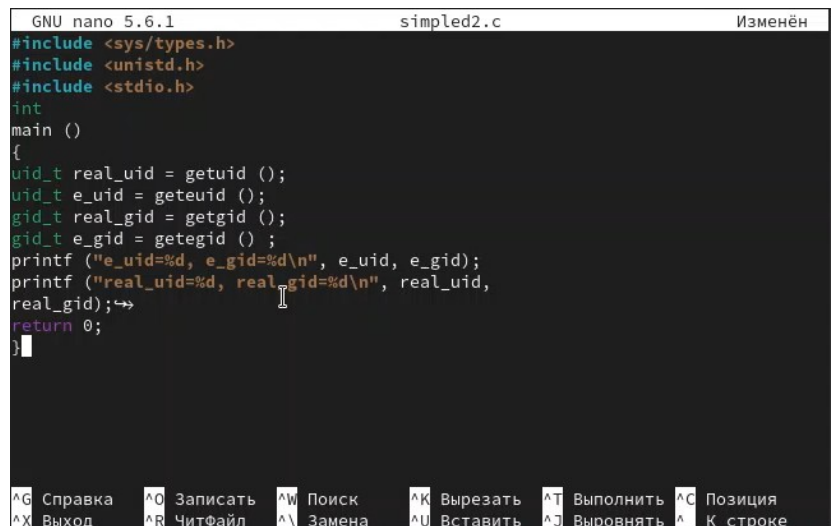
Компилирую файл, проверяю, запускаю, узнаю id



```
[guest@vparbatova ~]$ gcc simplified.c -o simplified
[guest@vparbatova ~]$ ls
simplified.c  Документы  Изображения  Общедоступные  Шаблоны
simplified  Видео      Загрузки     Музыка         'Рабочий стол'
[guest@vparbatova ~]$ ./simplified
uid=1001, gid=1001
[guest@vparbatova ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@vparbatova ~]$
```

Рис. 3.4: Работа с файлом

Текст второго файла

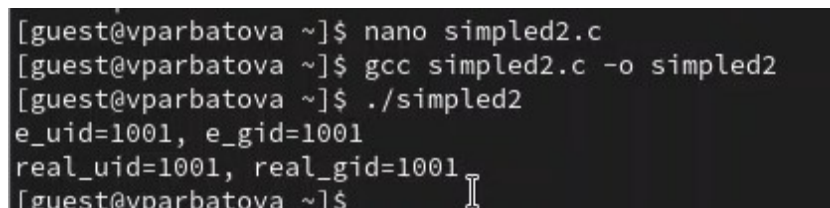


```
GNU nano 5.6.1 simplified2.c Изменён
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main ()
{
    uid_t real_uid = getuid ();
    uid_t e_uid = geteuid ();
    gid_t real_gid = getgid ();
    gid_t e_gid = getegid ();
    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
    printf ("real_uid=%d, real_gid=%d\n", real_uid,
    real_gid);
    return 0;
}
```

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^_ Замена ^U Вставить ^J Выровнять ^_ К строке

Рис. 3.5: Текст файла

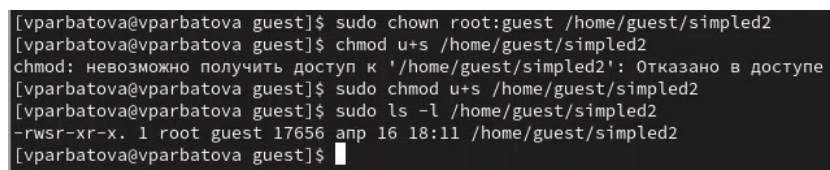
Повторяю операции со вторым файлом



```
[guest@vparbatova ~]$ nano simplified2
[guest@vparbatova ~]$ gcc simplified2.c -o simplified2
[guest@vparbatova ~]$ ./simplified2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@vparbatova ~]$
```

Рис. 3.6: Работа с файлом

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



```
[vparbatova@vparbatova guest]$ sudo chown root:guest /home/guest/simplified2
[vparbatova@vparbatova guest]$ chmod u+s /home/guest/simplified2
chmod: невозможно получить доступ к '/home/guest/simplified2': Отказано в доступе
[vparbatova@vparbatova guest]$ sudo chmod u+s /home/guest/simplified2
[vparbatova@vparbatova guest]$ sudo ls -l /home/guest/simplified2
-rwsr-xr-x. 1 root guest 17656 anp 16 18:11 /home/guest/simplified2
[vparbatova@vparbatova guest]$
```

Рис. 3.7: Изменение прав доступа

Сравниваю выводы, моя команда вывела меньше информации

```
[vparbatova@vparbatova guest]$ sudo /home/guest/simplified2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[vparbatova@vparbatova guest]$ id
uid=1000(vparbatova) gid=1000(vparbatova) группы=1000(vparbatova),10(wheel) конт
екст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[vparbatova@vparbatova guest]$ sudo id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconf
ined_t:s0-s0:c0.c1023
```

Рис. 3.8: Сравнение выводов

Текст файла



```
GNU nano 5.6.1 readfile.c Изменён
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;
}
```

[^]G Справка [^]O Записать [^]W Поиск [^]K Вырезать [^]T Выполнить [^]C Позиция
[^]X Выход [^]R ЧитФайл [^]\ Замена [^]U Вставить [^]J Выводить [^]_ К строке

Рис. 3.9: Текст файла

Создаю файл, открываю его в редакторе, компилирую, проверяю

```
[guest@vparbatova ~]$ touch readfile.c
[guest@vparbatova ~]$ nano readfile.c
[guest@vparbatova ~]$ gcc readfile.c -o readfile
[guest@vparbatova ~]$ ls
dir1      simplified  simplified.c  Загрузки  Общедоступные
readfile  simplified2  Видео        Изображения  'Рабочий стол'
readfile.c  simplified2.c  Документы  Музыка      Шаблоны
```

Рис. 3.10: Подготовка файла

Продолжаю изменять права от имени суперпользователя

```
[vparbatova@vparbatova guest]$ sudo chown root:guest /home/guest/readfile.c
[vparbatova@vparbatova guest]$ chmod u+s /home/guest/readfile.c
chmod: невозможно получить доступ к '/home/guest/readfile.c': Отказано в доступе
[vparbatova@vparbatova guest]$ sudo chmod u+s /home/guest/readfile.c
[vparbatova@vparbatova guest]$ sudo chmod 700 /home/guest/readfile.c
[vparbatova@vparbatova guest]$ sudo chmod -r /home/guest/readfile.c
[vparbatova@vparbatova guest]$ sudo chmod u+s /home/guest/readfile.c
```

Рис. 3.11: Изменение прав

Пытаюсь от имени пользователя guest прочитать файл, не получается

```
[guest@vparbatova ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@vparbatova ~]$
```

Рис. 3.12: Попытка прочесть файл

Пытаюсь прочесть файл с помощью нашего файла, получаю отказ в доступе

```
[guest@vparbatova ~]$ ./readfile readfile.c
```

Рис. 3.13: Попытка прочесть файл

Пытаюсь прочесть другой файл

[illegible]

Рис. 3.14: Попытка прочесть файл

Пробуем прочесть эти же файлы от имени суперпользователя и чтение файлов проходит успешно

```
[vparbatova@vparbatova guest]$ sudo /home/guest/readfile /etc/shadow
root:$6$cE9w5PKT13nMdw0/$mqnvxyg5cKjI5kIvn9MAFX2.H0CcZrXZo886eEfHi2kQ2LexRdSruI
Zt08.84iBrG94PbBzrk3Qk3bDeEZH30::0:99999:7:::
bin:!:19820:0:99999:7:::
daemon:!:19820:0:99999:7:::
adm:!:19820:0:99999:7:::
lp:!:19820:0:99999:7:::
sync:!:19820:0:99999:7:::
shutdown:!:19820:0:99999:7:::
```

Рис. 3.15: Чтение файлов

Проверяем папку tmp на наличие атрибута Sticky, т.к. в выводе есть буква t, то атрибут установлен

```
[vparbatova@vparbatova guest]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 anp 16 18:27 tmp
```

Рис. 3.16: Атрибут установлен

От имени пользователя guest создаю файл с текстом, добавляю права на чтение и запись для других пользователей

```
[guest@vparbatova ~]$ echo "text" > /tmp/file01.txt
[guest@vparbatova ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 anp 16 18:29 /tmp/file01.txt
[guest@vparbatova ~]$ chmod o+rw /tmp/file01.txt
[guest@vparbatova ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 anp 16 18:29 /tmp/file01.txt
[guest@vparbatova ~]$
```

Рис. 3.17: Изменение прав

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. Далее пробуем удалить файл, снова получаем отказ.

```
[guest@vparbatova ~]$ su guest2
Пароль:
[guest2@vparbatova guest]$ cat /tmp/file01.txt
text
[guest2@vparbatova guest]$ echo "text2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vparbatova guest]$ echo "text3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vparbatova guest]$ cat /tmp/file01.txt
text
[guest2@vparbatova guest]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
```

Рис. 3.18: Эксперименты

От имени суперпользователя снимаем с директории атрибут Sticky

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

Рис. 3.19: Снятие атрибута

Проверяем, что атрибут действительно снят

```
[guest2@vparbatova guest]$ ls -l / | grep tmp
drwxrwxrwx. 15 root root 4096 апр 16 18:33 tmp
```

Рис. 3.20: Проверка

Далее был выполнен повтор предыдущих действий. По результатам без Sticky-бита запись в файл и дозапись в файл осталась невозможной, зато удаление файла прошло успешно

```

[guest2@vparbatova guest]$ cat /tmp/file01.txt
text
[guest2@vparbatova guest]$ echo "text2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vparbatova guest]$ cat /tmp/file01.txt
text
[guest2@vparbatova guest]$ echo "text3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@vparbatova guest]$ rm /tmp/file01.txt
rm: удалить защищённый от записи обычный файл '/tmp/file01.txt'? y
[guest2@vparbatova guest]$ ls -l / | grep tmp
drwxrwxrwx. 15 root root 4096 апр 16 18:35 tmp
[guest2@vparbatova guest]$ ls -l
ls: невозможно получить доступ к '-': Нет такого файла или каталога
ls: невозможно получить доступ к 'l': Нет такого файла или каталога
[guest2@vparbatova guest]$ ls -l
итого 72
drwxrwxrwx. 2 guest guest 19 апр 2 17:03 dir1
-rwxr-xr-x. 1 guest guest 17600 апр 16 18:22 readfile
--ws----- 1 root guest 402 апр 16 18:22 readfile.c
-rwxr-xr-x. 1 guest guest 17552 апр 16 18:07 simplified
-rwsr-xr-x. 1 root guest 17656 апр 16 18:11 simplified2
-rw-r--r--. 1 guest guest 303 апр 16 18:11 simplified2.c
-rw-r--r--. 1 guest guest 175 апр 16 18:05 simplified.c
drwxr-xr-x. 2 guest guest 6 фев 19 16:19 Видео
drwxr-xr-x. 2 guest guest 6 фев 19 16:19 Документы
drwxr-xr-x. 2 guest guest 6 фев 19 16:19 Загрузки
drwxr-xr-x. 2 guest guest 6 фев 19 16:19 Изображения
drwxr-xr-x. 2 guest guest 6 фев 19 16:19 Музыка
drwxr-xr-x. 2 guest guest 6 фев 19 16:19 Общедоступные
drwxr-xr-x. 2 guest guest 6 фев 19 16:19 'Рабочий стол'

```

Рис. 3.21: Эксперименты 2

4 Выводы

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

Список литературы