

Лабораторная работа №5

Основы информационной безопасности

Феоктистов Владислав Сергеевич

22 сентября 2022

Российский университет дружбы народов, Москва, Россия

НПМбд-01-19

Целью данной работы является:

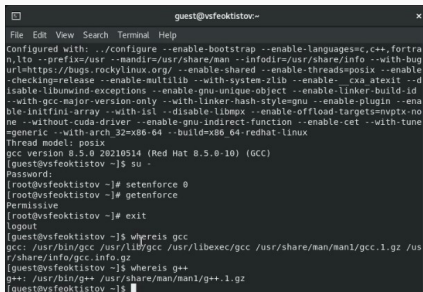
- изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов;
- получение практических навыков работы в консоли с дополнительными атрибутами;
- рассмотрение работы механизма смены идентификаторов процессов пользователей;
- изучение влияния Sticky-бита на запись и удаление файлов.

- Создать программу, выводящую реальные и эффективные идентификторы (uid, gid), посмотреть и сравнить результаты с выводом команды `id` до и после добавления SetUID- и SetGID-битов;
- Создать программу для чтения содержимого текстовых файлов и проверить его работу чтения файла с разрешением на его чтение только для владельца и с установленным SetUID-битом, запуская программу от другого пользователя, не являющегося владельцем файла. Сравнить с использованием команды `cat`;
- Изучить влияние Sticky-бита на работу с файлами и возможностью их удаления, работая с каталогом `/tmp`.

Ход выполнения лабораторной работы

Подготовка лабораторного стенда

Перед выполнением лабораторной работы необходимо убедиться, что установлен компилятор gcc и отключен механизм защиты для работы со Sticky-битом.

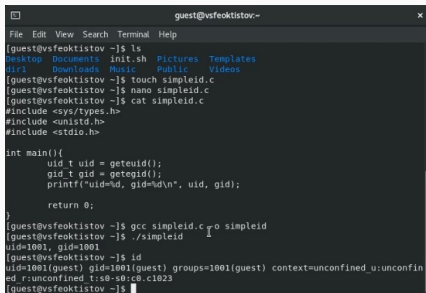


```
guest@vsfeektistov:~  
File Edit View Search Terminal Help  
Configured with: ../configure --enable-bootstrap --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bug-url=https://bugs.rockylinux.org/ --enable-shared --enable-threads=posix --enable-checking=release --enable-multilib --with-system-zlib --enable-cxx-exceptions --enable-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-lsl --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  
Thread model: posix  
gcc version 8.5.0 20210514 (Red Hat 8.5.0-10) (GCC)  
[guest@vsfeektistov ~]$ su -  
Password:  
[root@vsfeektistov ~]# setenforce 0  
[root@vsfeektistov ~]# getenforce  
Permissive  
[root@vsfeektistov ~]# exit  
logout  
[guest@vsfeektistov ~]$ 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  
[guest@vsfeektistov ~]$ whereis g++  
g++: /usr/bin/g++ /usr/share/man/man1/g++.1.gz  
[guest@vsfeektistov ~]$
```

Figure 1: Подготовка лабораторного стенда

Написание первой версии программы вывода uid и gid

- От имени пользователя *guest* создали программу *simpleid.c*, которая выводит эффективные идентификаторы uid и gid;
- Скомпилировали программу *simpleid.c* в исполняемый файл *simpleid* и запустим его;
- Выводимые эффективные идентификаторы совпадали с идентификаторами пользователя, который их запускал.

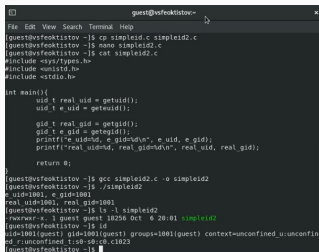


```
guest@vsfeoktistov:~  
File Edit View Search Terminal Help  
[guest@vsfeoktistov ~]$ ls  
Desktop Documents init.sh Pictures Templates  
dir1 Downloads Music Public Videos  
[guest@vsfeoktistov ~]$ touch simpleid.c  
[guest@vsfeoktistov ~]$ nano simpleid.c  
[guest@vsfeoktistov ~]$ cat simpleid.c  
#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@vsfeoktistov ~]$ gcc simpleid.c -o simpleid  
[guest@vsfeoktistov ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@vsfeoktistov ~]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@vsfeoktistov ~]$
```

Figure 2: Первая версия программы

Написание второй версии программы вывода uid и gid

- От того же пользователя скопировали файл *simpleid.c* в файл *simpleid2.c* и добавили в программе вывод реальных идентификаторов;
- Снова скомпилировали файл и запустили его;
- Выводимые идентификаторы (реальные и эффективные) совпали с идентификаторами команды *id*, как и до этого.



```
quest@vsfeektistov:~$ cp simpleid.c simpleid2.c
quest@vsfeektistov:~$ nano simpleid2.c
quest@vsfeektistov:~$ 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();

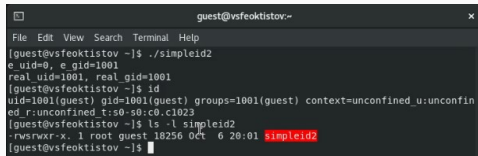
    gid_t real_gid = getgid();
    gid_t e_gid = getegid();
    printf("e uid=%d, e gid=%d\n", e_uid, e_gid);
    printf("real uid=%d, real gid=%d\n", real_uid, real_gid);

    return 0;
}
quest@vsfeektistov:~$ gcc simpleid2.c -o simpleid2
quest@vsfeektistov:~$ ./simpleid2
e uid=1001, e gid=1001
real uid=1001, real gid=1001
quest@vsfeektistov:~$ ls -l simpleid2
-rwxr-xr-x 1 quest quest 10256 Oct 6 20:01 ./simpleid2
quest@vsfeektistov:~$ id
uid=1001(quest) gid=1001(quest) groups=1001(quest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
quest@vsfeektistov:~$
```

Figure 3: Вторая версия программы

Запуск программы после смены владельца и установки SetUID-бита

- Через root-пользователя поменяли владельца исполняемого файла и устанавливаем ему SetUID-бит;
- После запуска программы от имени пользователя *guest*, эффективный идентификатор `uid` отличался от реального `uid` и `uid` команды `id`. Идентификатор `gid` напротив же не изменился.

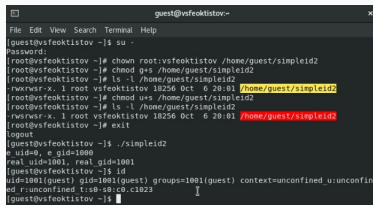


```
guest@vsfeoktistov:~  
File Edit View Search Terminal Help  
[guest@vsfeoktistov ~]$ ./simpleid2  
e_uid=0, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@vsfeoktistov ~]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@vsfeoktistov ~]$ ls -l simpleid2  
-rwsrwxr-x. 1 root guest 10256 Oct  6 20:01 simpleid2  
[guest@vsfeoktistov ~]$
```

Figure 4: Работа программы после смены ее владельца и установки SetUID

Запуск программы после смены группы владельцев и установки SetGID-бита

- Через root-пользователя поменяли группу владельцев исполняемого файла и устанавливаем ему SetGID-бит (заодно оставили предыдущего владельца и SetUID-бит);
- После запуска программы от имени пользователя *guest*, уже и эффективный идентификатор gid отличался от реального gid и gid команды *id*.



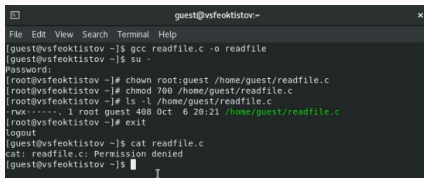
```
guest@vsfeoktistov:~  
File Edit View Search Terminal Help  
[guest@vsfeoktistov ~]$ su -  
Password:  
[root@vsfeoktistov ~]# chown root:vsfeoktistov /home/guest/simpleid2  
[root@vsfeoktistov ~]# chmod g+s /home/guest/simpleid2  
[root@vsfeoktistov ~]# ls -l /home/guest/simpleid2  
-rwxrwsr-x. 1 root vsfeoktistov 18256 Oct  6 20:01 /home/guest/simpleid2  
[root@vsfeoktistov ~]# chmod u+s /home/guest/simpleid2  
[root@vsfeoktistov ~]# ls -l /home/guest/simpleid2  
-rwxrwsr-x. 1 root vsfeoktistov 18256 Oct  6 20:01 /home/guest/simpleid2  
[root@vsfeoktistov ~]# exit  
logout  
[guest@vsfeoktistov ~]$ ./simpleid2  
e_uid=0, e_gid=1000  
real_uid=1001, real_gid=1001  
[guest@vsfeoktistov ~]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@vsfeoktistov ~]$
```

Figure 5: Работа программы после смены группы владельцев и установки SetGID

Вывод: реальные идентификаторы показывают идентификаторы пользователя, который запустил программу, а эффективные - идентификатор владельца файла.

Программа вывода содержимого файлов

- Написали программу *readfile.c* для вывода в консоль содержимого указанного файла и скомпилировали её;
- Через root-пользователя поменяли владельца исходного файла и разрешили его чтение только новому владельцу;
- Попробовали через пользователя *guest* прочитать его содержимое.

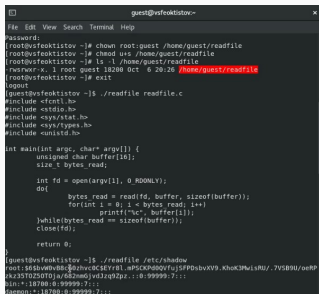


```
guest@vsfeoktistov ~  
File Edit View Search Terminal Help  
[guest@vsfeoktistov ~]$ gcc readfile.c -o readfile  
[guest@vsfeoktistov ~]$ su -  
Password:  
[root@vsfeoktistov ~]# chown root:guest /home/guest/readfile.c  
[root@vsfeoktistov ~]# chmod 700 /home/guest/readfile.c  
[root@vsfeoktistov ~]# ls -l /home/guest/readfile.c  
-rwx----- 1 root guest 408 Oct  6 20:21 /home/guest/readfile.c  
[root@vsfeoktistov ~]# exit  
logout  
[guest@vsfeoktistov ~]$ cat readfile.c  
cat: readfile.c: Permission denied  
[guest@vsfeoktistov ~]$
```

Figure 6: Программа вывода содержимого файлов

Программа вывода содержимого файлов со SetUID-битом

- Добавили через root-пользователя SetUID-бит для исполняемого файла, установив ему того же владельца, что и у исходного файла;
- Попробовали прочитать содержимое исходного файла и файла `/etc/shadow`.

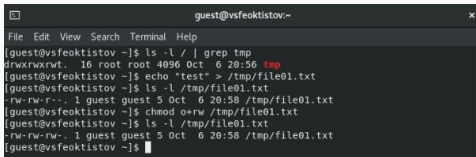


```
guest@vsfeaktistov:~$  
File Edit View Search Terminal Help  
Password:  
[root@vsfeaktistov ~]# chown root:guest /home/guest/readfile  
[root@vsfeaktistov ~]# chmod u+s /home/guest/readfile  
[root@vsfeaktistov ~]# ls -l /home/guest/readfile  
-rwxrwx--x. 1 root guest 18200 Oct 6 20:26 /home/guest/readfile  
[root@vsfeaktistov ~]# exit  
logout  
[guest@vsfeaktistov ~]$ ./readfile 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[10];  
    size_t bytes_read;  
  
    int fd = open(argv[1], O_RDONLY);  
    do {  
        bytes_read = read(fd, buffer, sizeof(buffer));  
        for(int i = 0; i < bytes_read; i++)  
            printf("%c", buffer[i]);  
    } while(bytes_read == sizeof(buffer));  
    close(fd);  
  
    return 0;  
}  
  
[guest@vsfeaktistov ~]$ ./readfile /etc/shadow  
root:$6$bv8v88$6zhvc0CSEY8l.mPCKK40Q/fujSPDbvXV9.KhoK3MwlsRU/.7VSB9U/oeRP  
ckz35T0Z50T0ja/662mm6jvd3q9Zpz.:0:99999:7:::  
bin:*:18700:0:99999:7:::  
daemon:*:18700:0:99999:7:::
```

Figure 7: Программа вывода содержимого файлов со SetUID-битом

Проверка Sticky-бита и создание файла в каталоге

- Посмотрели наличие Sticky-бита у каталога `/tmp`;
- Создали в нем файл `file01.txt` с сообщением `test` и добавили права чтения и записи для группы “остальные пользователи”.

A terminal window titled 'guest@vsfeoktistov:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
[guest@vsfeoktistov ~]$ ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 Oct  6 20:56 tmp
[guest@vsfeoktistov ~]$ echo "test" > /tmp/file01.txt
[guest@vsfeoktistov ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Oct  6 20:58 /tmp/file01.txt
[guest@vsfeoktistov ~]$ chmod o+rw /tmp/file01.txt
[guest@vsfeoktistov ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 Oct  6 20:58 /tmp/file01.txt
[guest@vsfeoktistov ~]$
```

Figure 8: Проверка Sticky-бита и создание файла в каталоге

Работа с текстовым файлом при наличии Sticky-бита

Попробовали произвести чтение, дозапись, перезапись и удаление текстового файла при наличии Sticky-бита у каталога, в котором находится этот файл (*/tmp*). Из картинки ниже, видно, что невозможно было только удаление.

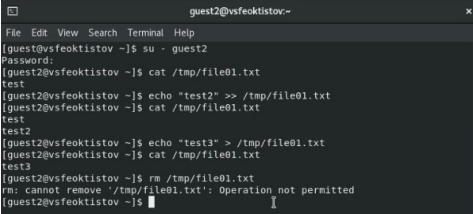
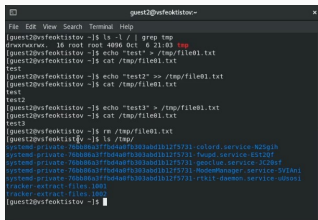
A terminal window titled 'guest2@vsfeoktistov:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows a sequence of commands and their outputs. First, 'su - guest2' is run. Then, 'cat /tmp/file01.txt' shows 'test'. Next, 'echo "test2" >> /tmp/file01.txt' is run. Then, 'cat /tmp/file01.txt' shows 'test' and 'test2'. Next, 'echo "test3" > /tmp/file01.txt' is run. Then, 'cat /tmp/file01.txt' shows 'test3'. Finally, 'rm /tmp/file01.txt' is run, resulting in the error message 'rm: cannot remove '/tmp/file01.txt': Operation not permitted'.

Figure 9: Работа с текстовым файлом при наличии Sticky-бита

Работа с текстовым фалом при отсутствии Sticky-бита

Повторили те же действия, только уже после снятия Sticky-бита с каталога `/tmp` [**cmd:** `chmod -t /tmp`] от имени суперпользователя. В результате, успешно сработали все команды, в том числе и удаление.



```
guest2@vsfeektistov~  
File Edit View Search Terminal Help  
[guest2@vsfeektistov ~]$ ls -l / | grep tmp  
drwxrwxrwx. 16 root root 4096 Oct 6 21:03 tmp  
[guest2@vsfeektistov ~]$ echo "test" > /tmp/file01.txt  
[guest2@vsfeektistov ~]$ cat /tmp/file01.txt  
test  
[guest2@vsfeektistov ~]$ echo "test2" >> /tmp/file01.txt  
[guest2@vsfeektistov ~]$ cat /tmp/file01.txt  
test  
test2  
[guest2@vsfeektistov ~]$ echo "test3" > /tmp/file01.txt  
[guest2@vsfeektistov ~]$ cat /tmp/file01.txt  
test3  
[guest2@vsfeektistov ~]$ rm /tmp/file01.txt  
[guest2@vsfeektistov ~]$ ls /tmp/  
systemd-private-76ab8ba3fbd4aefb383abd1b1275731-color.service-N25qih  
systemd-private-76ab8ba3fbd4aefb383abd1b1275731-ftp.service-E5T2qf  
systemd-private-76ab8ba3fbd4aefb383abd1b1275731-gssd.service-XZ28st  
systemd-private-76ab8ba3fbd4aefb383abd1b1275731-ModemManager.service-5vIAm1  
systemd-private-76ab8ba3fbd4aefb383abd1b1275731-rtkit-daemon.service-uWnos1  
tracker-extract-files-1601  
tracker-extract-files-1602  
[guest2@vsfeektistov ~]$
```

Figure 10: Работа с текстовым фалом при отсутствии Sticky-бита

Вывод: Sticky-бит запрещает удаление содержимого каталога для пользователей, не являющихся владельцем директории. Остальные действия разрешены в соответствии с установленными на них правами.

В процессе выполнения лабораторной работы:

- изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов;
- получили практические навыки работы в консоли с дополнительными атрибутами;
- рассмотрели работу механизма смены идентификаторов процессов пользователей;
- изучили влияние Sticky-бита на запись и удаление файлов.