

Л.5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Греков Максим Сергеевич

2021

RUDN University, Moscow, Russian Federation

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.

Подготовка лабораторного стенда

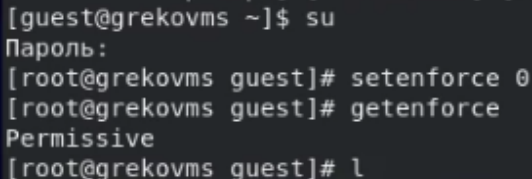
При подготовке стенда убедились, что в системе установлен компилятор gcc (для этого ввели команду `gcc -v`):

```
[guest@grekovms ~]$ 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 --w
ith-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-li
nker-build-id --with-gcc-major-version-only --with-linker-hash-style=gnu --enabl
e-plugin --enable-initfini-array --with-isl --disable-libmpx --enable-offload-ta
rgets=nvptx-none --without-cuda-driver --enable-gnu-indirect-function --enable-c
et --with-tune=generic --with-arch_32=x86_64 --build=x86_64-redhat-linux
Модель многопоточности: posix
gcc версия 8.4.1 20200928 (Red Hat 8.4.1-1) (GCC)
[guest@grekovms ~]$
```

Figure 1: Компилятор gcc

Отключили систему запретов до очередной перезагрузки системы командой `setenforce 0`

После этого команда `getenforce` вывела `Permissive`.

A terminal window with a dark background and light-colored text. The text shows a user switching from 'guest' to 'root' using 'su', then running 'setenforce 0' and 'getenforce' which returns 'Permissive', and finally running 'l' (likely 'ls').


```
[guest@grekovms ~]$ su
Пароль:
[root@grekovms guest]# setenforce 0
[root@grekovms guest]# getenforce
Permissive
[root@grekovms guest]# l
```

Figure 2: Система запретов

Ход работы

Программа simpleid.c

Вошли в систему от имени пользователя guest и создали программу simpleid.c:

```
[guest@grekovms ~]$ touch simpleid.c
[guest@grekovms ~]$ 

#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;
}
```

Figure 3: Программа simpleid.c

Скомпилировали программу и убедились, что файл программы создан:

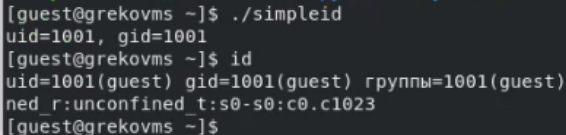
```
gcc simpleid.c -o simpleid
```

```
[guest@grekovms ~]$ gcc simpleid.c -o simpleid
[guest@grekovms ~]$ ls
dir1      simpleid.c  Документы  Изображения
simpleid   Видео      Загрузки   Музыка
```

Figure 4: Компиляция simpleid.c

Выполнение simpleid.c и id

Выполнили программу simpleid: ./simpleid, а также системную программу id:

A terminal window with a dark background and light-colored text. The prompt is [guest@grekovms ~]\$. The first command is ./simpleid, which outputs uid=1001, gid=1001. The second command is id, which outputs uid=1001(guest) gid=1001(guest) группы=1001(guest) ned_r:unconfined_t:s0-s0:c0.c1023. The prompt returns to [guest@grekovms ~]\$.

```
[guest@grekovms ~]$ ./simpleid
uid=1001, gid=1001
[guest@grekovms ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest)
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@grekovms ~]$
```

Figure 5: Выполнение simpleid.c и id

Получили идентичные результаты рассматриваемых параметров

Программа simpleid2.c

Усложнили программу, добавив вывод действительных идентификаторов, и назвали её 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;
}
```

Figure 6: Программа simpleid2.c

Компиляция и выполнение simpleid2.c

Скомпилировали и запустили программу simpleid2.c:

```
[guest@grekovms ~]$ touch simpleid2.c
[guest@grekovms ~]$ gcc simpleid2.c -o simpleid2
[guest@grekovms ~]$ ls
dir1      simpleid2.c  Документы  Музыка
simpleid   simpleid.c   Загрузки   Общедоступные
simpleid2  Видео       Изображения 'Рабочий стол'
[guest@grekovms ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@grekovms ~]$
```

Figure 7: Компиляция и выполнение simpleid2.c

Теперь видим не только текущих группу и пользователя, но и владельца файла.

Смена владельца и атрибут s

От имени суперпользователя выполнили команды:

```
chown root:guest /home/guest/simpleid2
```

```
chmod u+s /home/guest/simpleid2
```

Тем самым, сменили владельца файла и добавили ему дополнительный атрибут (SetUID).

Затем выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

```
ls -l simpleid2
```

Смена владельца и атрибут s

```
пароль:
[root@grekovms guest]# sudo chown root:guest /home/guest/simpleid2
[root@grekovms guest]# ls -l simpleid2
-rwxrwxr-x. 1 root guest 17648 ноя 12 22:31 simpleid2
[root@grekovms guest]# chmod u+s /home/guest/simpleid2
[root@grekovms guest]# ls -l
итого 48
drwxrwxrwx. 2 guest guest 19 окт 29 21:03 dir1
-rwxrwxr-x. 1 guest guest 17544 ноя 12 22:25 simpleid
-rwsrwxr-x. 1 root guest 17648 ноя 12 22:31 simpleid2
```

Figure 8: Смена владельца и атрибут s

Запуск simpleid2 с SetUID

Запустили simpleid2 и id, вновь получили идентичные результаты.

Убедились в принадлежности файла пользователю root.

```
simpleid2
[root@grekovms guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@grekovms guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=
ined_t:s0-s0:c0.c1023
[root@grekovms guest]#
```

Figure 9: Запуск simpleid2 с SetUID

Запуск simpleid2 с SetGID

Проделали то же самое относительно SetGID-бита:

```
[root@grekovms guest]# ls -l simpleid2
-rwxrwsr-x. 1 root guest 17648 ноя 12 22:31 simpleid2
[root@grekovms guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@grekovms guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined
ined t:s0-s0:c0.c1023
[root@grekovms guest]# l
```

Figure 10: Запуск simpleid2 с SetGID

Создали 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;
}
```

Figure 11: Программа readfile.c

Откомпилировали программу и изменили владельца у файла `readfile.c` и права так, чтобы только суперпользователь (`root`) мог прочесть его, а `guest` не мог, убедились в правильности, получив отказ в доступе:

```
[root@grekovms guest]# sudo chown guest2:guest /home/guest/readfile.c
[root@grekovms guest]# ls -l readfile.c
-rw-rw-rw-. 1 guest2 guest 408 ноя 12 23:01 readfile.c
[root@grekovms guest]# chmod ug-r /home/guest/readfile.c
[root@grekovms guest]# ls -l readfile.c
--w--w-r--. 1 guest2 guest 408 ноя 12 23:01 readfile.c
[guest@grekovms ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@grekovms ~]$
```

Figure 12: Компиляция и изменение `readfile.c`

Изменение владельца readfile с SetUID

Сменили у программы readfile владельца и установили SetUID-бит:

```
[root@grekovms guest]# sudo chown guest2:guest /home/guest/readfile
[root@grekovms guest]# ls -l readfile
-rwxr-xr-x. 1 guest2 guest 17592 ноя 12 23:02 readfile
[root@grekovms guest]# sudo chmod u+s /home/guest/readfile
[root@grekovms guest]# ls -l readfile
-rwsr-xr-x. 1 guest2 guest 17592 ноя 12 23:02 readfile
[root@grekovms guest]#
```

Figure 13: Изменение владельца readfile с SetUID

Попытка прочтения

Проверили, может ли программа readfile прочитать файлы readfile.c и /etc/shadow, в обоих случаях это не удалось, потому что владелец файла программы guest2:

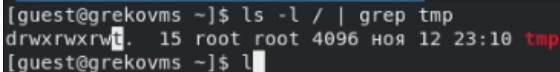
```
az=38;5;9:*.lha=38;5;9:*.lz4=38;5;9:*.lzh=38;5;9:*.lzma=38;5;9:*.tlz=38;5;9:*.tx
z=38;5;9:*.tzo=38;5;9:*.t7z=38;5;9:*.zip=38;5;9:*.z=38;5;9:*.dz=38;5;9:*.gz=38;5
;9:*.lrz=38;5;9:*.lz=38;5;9:*.lzo=38;5;9:*.xz=38;5;9:*.zst=38;5;9:*.tztst=38;5;9
:*.bz2=38;5;9:*.bz=38;5;9:*.tbz=38;5;9:*.tbz2=38;5;9:*.tz=38;5;9:*.deb=38;5;9:*.r
pm=38;5;9:*.jar=38;5;9:*.war=38;5;9:*.ear=38;5;9:*.sar=38;5;9:*.rar=38;5;9:*.alz
=38;5;9:*.ace=38;5;9:*.zoo=38;5;9:*.cpio=38;5;9:*.7z=38;5;9:*.rz=38;5;9:*.cab=38
;5;9:*.wim=38;5;9:*.swm=38;5;9:*.dwm=38;5;9:*.esd=38;5;9:*.jpg=38;5;13:*.jpeg=38
;5;13:*.mjpg=38;5;13:*.mjpeg=38;5;13:*.gif=38;5;13:*.bmp=38;5;13:*.pbm=38;5;13:*.
pgm=38;5;13:*.ppm=38;5;13:*.tga=38;5;13:*.xbm=38;5;13:*.xpm=38;5;13:*.tif=38;5;
13:*.tiff=38;5;13:*.png=38;5;13:*.svg=38;5;13:*.svgz=38;5;13:*.mng=38;5;13:*.pcx
=38;5;13:*.mov=38;5;13:*.mpg=38;5;13:*.mpeg=38;5;13:*.m2v=38;5;13:*.mkv=38;5;13:
*.webm=38;5;13:*.ogm=38;5;13:*.mp4=38;5;13:*.m4v=38;5;13:*.mp4v=38;5;13:*.vob=38
;5;13:*.qt=38;5;13:*.nuv=38;5;13:*.wmv=38;5;13:*.asf=38;5;13:*.rm=38;5;13:*.rmvb
=38;5;13:*.flc=38;5;13:*.avi=38;5;13:*.fli=38;5;13:*.flv=38;5;13:*.gl=38;5;13:*.
dl=38;5;13:*.xcf=38;5;13:*.xwd=38;5;13:*.yuv=38;5;13:*.cgm=38;5;13:*.emf=38;5;13
:*.ogv=38;5;13:*.ogx=38;5;13:*.aac=38;5;45:*.au=38;5;45:*.flac=38;5;45:*.m4a=38;
5;45:*.mid=38;5;45:*.midi=38;5;45:*.mka=38;5;45:*.mp3=38;5;45:*.mpc=38;5;45:*.og
g=38;5;45:*.ra=38;5;45:*.wav=38;5;45:*.oga=38;5;45:*.opus=38;5;45:*.spx=38;5;45:
*.xspf=38;5;45:XDG_MENU_PREFIX=gnome-LANG=ru RU.UTF-8GDM LANG=ru RU.UTF-8HISTCON
TROL=ignoredupsDISPLAY=HOSTNAME=grevkoms.localdomainCOLORTERM=truecolorUSERNAM
E=guestXDG_VTNR=3SSH_AUTH_SOCK=/run/user/1001/keyring/sshXDG_SESSION_ID=5USER=gu
estDESKTOP_SESSION=gnomeWAYLAND_DISPLAY=wayland-@GNOME_TERMINAL_SCREEN=/org/gnom
e/terminal/screen/ac41a249_ade7_46a8_bb65_16cad02c47b3PWD=/home/guestSSH_ASKPASS
=/usr/libexec/openssh/gnome-ssh-askpassHOME=/home/guestXDG_SESSION_TYPE=waylandX
DG_DATA_DIRS=/home/guest:/local/share/flatpak/exports/share:/var/lib/flatpak/exp
orts/share:/usr/local/share:/usr/shareXDG_SESSION_DESKTOP=gnomeGJS_DEBUG_Ошибка
сегментирования
[guest@grevkoms ~]$ ./readfile
```

Figure 14: Попытка прочтения

Атрибут Sticky на /tmp

Выяснили, что на директории /tmp установлен атрибут Sticky:

`ls -l / | grep tmp`

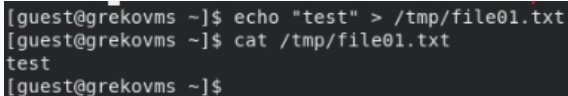


```
[guest@grekovms ~]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 ноя 12 23:10 tmp
[guest@grekovms ~]$ l
```

Figure 15: Атрибут Sticky на /tmp

От имени пользователя guest создали файл file01.txt в директории /tmp со словом test:

```
echo "test" > /tmp/file01.txt
```

A terminal window with a dark background and light gray text. It shows a user named 'guest' at a machine named 'grekovms' in the home directory '~'. The user enters the command 'echo "test" > /tmp/file01.txt' to create a file. Then, they enter 'cat /tmp/file01.txt' to verify its contents, which shows 'test'. The prompt returns to '~\$'.

```
[guest@grekovms ~]$ echo "test" > /tmp/file01.txt  
[guest@grekovms ~]$ cat /tmp/file01.txt  
test  
[guest@grekovms ~]$
```

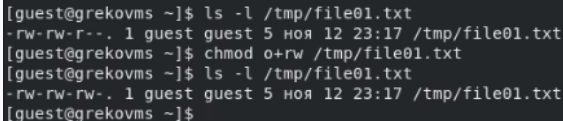
Figure 16: Создание file01.txt

Атрибуты file01.txt

Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные»:

```
chmod o+rw /tmp/file01.txt
```

```
ls -l /tmp/file01.txt
```



```
[guest@grekovms ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 12 23:17 /tmp/file01.txt
[guest@grekovms ~]$ chmod o+rw /tmp/file01.txt
[guest@grekovms ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 12 23:17 /tmp/file01.txt
[guest@grekovms ~]$
```

Figure 17: Атрибуты file01.txt

Работа с файлом file01.txt в директории с t

От пользователя guest2 (не являющегося владельцем) попробовали:

1. прочитать файл /tmp/file01.txt: `cat /tmp/file01.txt`
2. дозаписать в файл /tmp/file01.txt слово test2: `echo "test2" >> /tmp/file01.txt`
3. проверить содержимое файла: `cat /tmp/file01.txt`
4. записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию: `echo "test3" > /tmp/file01.txt`
5. проверить содержимое файла: `cat /tmp/file01.txt`
6. удалить файл /tmp/file01.txt: `rm /tmp/file01.txt`

Работа с файлом file01.txt в директории с t

Удалось дозаписать информацию в файл, перезаписать, прочесть, но не удалось удалить файл:

```
[guest2@grekovms guest]$ echo "test1" > /tmp/file01.txt
[guest2@grekovms guest]$ cat /tmp/file01.txt
test1
[guest2@grekovms guest]$ echo "test2" >> /tmp/file01.txt
[guest2@grekovms guest]$ cat /tmp/file01.txt
test1
test2
[guest2@grekovms guest]$ echo "test3" > /tmp/file01.txt
[guest2@grekovms guest]$ cat /tmp/file01.txt
test3
[guest2@grekovms guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@grekovms guest]$ l
```

Figure 18: Работа с файлом file01.txt в директории с t

Повысили свои права до суперпользователя командой `su -`

Выполнили после этого команду, снимающую атрибут t (Sticky-бит) с директории `/tmp`: `chmod -t /tmp`.

Покинули режим суперпользователя командой `exit`

От пользователя `guest2` проверили, что атрибута t у директории `/tmp` нет: `ls -l / | grep tmp`

Снятие атрибута t

```
[root@grekovms /]# chmod -t /tmp
[root@grekovms /]# ls -l / | grep tmp
drwxrwxrwx. 15 root root 4096 ноя 12 23:26 tmp
[root@grekovms /]# exit
выход
[guest@grekovms ~]$
```

Figure 19: Снятие атрибута t

Работа с файлом file01.txt в директории без t

Повторили все действия и, в отличие от предыдущего раза, теперь уже нам удалось удалить файл:

```
[guest2@grekovms tmp]$ cat file01.txt
test3
[guest2@grekovms tmp]$ echo "test2" >> file01.txt
[guest2@grekovms tmp]$ cat file01.txt
test3
test2
[guest2@grekovms tmp]$ echo "test3" > file01.txt
[guest2@grekovms tmp]$ cat file01.txt
test3
[guest2@grekovms tmp]$ rm file01.txt
[guest2@grekovms tmp]$ ls
systemd-private-05d8726b0d134d91a0b3aca8ca4925fe-color.service-zd35Vf
systemd-private-05d8726b0d134d91a0b3aca8ca4925fe-fwupd.service-wLj8G1
systemd-private-05d8726b0d134d91a0b3aca8ca4925fe-ModemManager.service-IY66Rf
systemd-private-05d8726b0d134d91a0b3aca8ca4925fe-rtkit-daemon.service-tloeDh
tracker-extract-files.1000
tracker-extract-files.1001
[guest2@grekovms tmp]$
```

Figure 20: Работа с файлом file01.txt в директории без t

Возврат атрибута t

Повысили свои права с помощью su, вернули директории /tmp атрибут t:

```
[guest@grekovms ~]$ su -  
Пароль:  
[root@grekovms ~]# chmod +t /tmp  
[root@grekovms ~]# ls -l / | grep tmp  
drwxrwxrwt. 16 root root 4096 ноя 12 23:31 tmp  
[root@grekovms ~]# exit  
выход  
[guest@grekovms ~]$
```

Figure 21: Возврат атрибута t

Вывод

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов.

Получили практические навыки работы в консоли с дополнительными атрибутами.

Рассмотрели работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

