

# Отчет по лабораторной работе №5

Дискреционное разграничение прав в Linux. Исследование влияния  
расширенных атрибутов

Евсеева Дарья Олеговна

8 октября, 2022

# Содержание

Цель работы	4
Задание	5
Теоретическое введение	6
Выполнение лабораторной работы	7
1. Работа с SetUID- и SetGID-битами . . . . .	7
2. Работа со Sticky-битом . . . . .	14
Выводы	17
Список литературы	18

## Список иллюстраций

1	Вход в систему от имени пользователя guest . . . . .	7
2	Создание файла . . . . .	7
3	Код программы . . . . .	8
4	Выполнение программы и команды id . . . . .	8
5	Создание файла . . . . .	8
6	Код программы . . . . .	9
7	Результаты запуска программы . . . . .	9
8	Установка атрибута s для пользователя . . . . .	10
9	Установка атрибута s для группы . . . . .	10
10	Создание файла . . . . .	11
11	Код программы . . . . .	11
12	Компиляция программы . . . . .	11
13	Смена владельца файла и изменение прав . . . . .	12
14	Смена владельца и установка SetUID-бита . . . . .	12
15	Чтение readfile.c от имени суперпользователя . . . . .	12
16	Чтение /etc/shadow от имени суперпользователя . . . . .	13
17	Чтение readfile.c от имени пользователя guest . . . . .	13
18	Чтение /etc/shadow от имени пользователя guest . . . . .	14
19	Проверка наличия атрибута Sticky . . . . .	14
20	Создание файла и изменение прав . . . . .	14
21	Чтение файла от имени пользователя guest2 . . . . .	15
22	Попытка дозаписи в файл . . . . .	15
23	Попытка перезаписи файла . . . . .	15
24	Попытка удаления файла . . . . .	15
25	Снятие атрибута Sticky . . . . .	15
26	Проверка атрибутов . . . . .	16
27	Попытка повторного выполнения команд . . . . .	16
28	Возвращение атрибута Sticky . . . . .	16

## Цель работы

Целью данной работы является изучение механизмов изменения идентификаторов и применения SetUID-, SetGID- 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.

## Задание

1. Провести работу с SetUID- и SetGID-битами.
2. Провести работу со Sticky-битом.

# Теоретическое введение

Дискреционное разграничение доступа — подход к разграничению доступа, предполагающий назначение владельцев объектов, которые по собственному усмотрению определяют права доступа субъектов (других пользователей) к объектам (файлам), которыми владеют.

Дискреционные механизмы разграничения доступа используются для разграничения прав доступа процессов как обычных пользователей, так и для ограничения прав системных программ (например, служб операционной системы), которые работают от лица псевдопользовательских учетных записей.

# Выполнение лабораторной работы

Выполнять работу будем в операционной системе, установленной при выполнении первой лабораторной работы.

## 1. Работа с SetUID- и SetGID-битами

Для начала войдем в систему от имени пользователя guest.

```
[doevseeva@doevseeva ~]$ su - guest
Password:
[guest@doevseeva ~]$
```

Рис. 1: Вход в систему от имени пользователя guest

Далее создадим файл simpleid.c и запишем в него код программы.

```
[guest@doevseeva ~]$ touch simpleid.c
[guest@doevseeva ~]$ vi simpleid.c
```

Рис. 2: Создание файла

```

#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;
}

```

"simpleid.c" 11L, 181B 11,0-1 All

Рис. 3: Код программы

Скомпилируем программу и выполним ее, сравнив результаты с выводом команды `id`.

```

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

```

Рис. 4: Выполнение программы и команды `id`

Мы видим, что результаты вывода программы и команды совпадают и соответствуют действительности.

Далее создадим файл `simpleid2.c` и запишем в него ранее написанную программу, добавив вывод действительных идентификаторов.

```

[guest@doevseeva ~]$ cp simpleid.c simpleid2.c
[guest@doevseeva ~]$ vi simpleid2.c

```

Рис. 5: Создание файла



```

#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;
}

```

"simpleid2.c" 16L, 313B 16,0-1 All

Рис. 6: Код программы

Скомпилируем и запустим программу.

```

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

```

Рис. 7: Результаты запуска программы

Здесь мы видим, что полученные значения для пар uid и gid совпадают.

Далее от имени суперпользователя сменим владельца файла simpleid2 и установим на него атрибут s для пользователя, после чего проверим правильность выполненных команд и сравним результат запуска simpleid2 с выводом команды id для суперпользователя и для пользователя guest.

```

[guest@doevseeva ~]$ su
Password:
[root@doevseeva guest]# chown root:guest /home/guest/simpleid2
[root@doevseeva guest]# chmod u+s /home/guest/simpleid2
[root@doevseeva guest]#
[root@doevseeva guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26008 Oct  7 23:56 simpleid2
[root@doevseeva guest]#
[root@doevseeva guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@doevseeva guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@doevseeva guest]# exit
exit
[guest@doevseeva ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@doevseeva ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@doevseeva ~]$
[guest@doevseeva ~]$

```

Рис. 8: Установка атрибута s для пользователя

Мы видим, что при выводе результатов от имени суперпользователя все результаты совпадают по значениям, однако при выводе результатов от имени пользователя guest можно заметить, что значение `e_uid` остается соответствующим суперпользователю, а остальные значения соответствуют значениям для пользователя guest.

Далее сделаем те же действия, установив на файл от имени суперпользователя атрибут s для группы.

```

[guest@doevseeva ~]$ su
Password:
[root@doevseeva guest]# chmod g+s /home/guest/simpleid2
[root@doevseeva guest]# ls -l simpleid2
-rwsr-xr-x. 1 root guest 26008 Oct  7 23:56 simpleid2
[root@doevseeva guest]#
[root@doevseeva guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@doevseeva guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@doevseeva guest]# exit
exit
[guest@doevseeva ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@doevseeva ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@doevseeva ~]$

```

Рис. 9: Установка атрибута s для группы

Здесь мы видим, что при выводе результатов от имени суперпользователя все результаты совпадают по значениям, кроме значения `g_uid`, которое соответствует

группе guest, а при выводе результатов от имени пользователя guest значение e\_uid остается соответствующим суперпользователю, а значение g\_uid с остальными результатами также соответствует группе и пользователю guest.

Теперь создадим программу readfile.c и откомпилируем ее.

```
[guest@doevseeva ~]$ touch readfile.c
[guest@doevseeva ~]$ vi readfile.c
```

Рис. 10: Создание файла

```
#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;
}

"readfile.c" 22L, 422B                               5,16      All
```

Рис. 11: Код программы

```
[guest@doevseeva ~]$ gcc readfile.c -o readfile
```

Рис. 12: Компиляция программы

Сменим владельца у файла readfile.c и изменим права так, чтобы только супер-пользователь мог прочитать его, а guest не мог, и проверим успешность изменения прав.

```
[guest@doevseeva ~]$ su
Password:
[root@doevseeva guest]# chown root:root /home/guest/readfile.c
[root@doevseeva guest]# chmod o-r /home/guest/readfile.c
[root@doevseeva guest]# chmod g-r /home/guest/readfile.c
[root@doevseeva guest]# ls -l /home/guest/
total 96
drwxrwx---. 2 guest guest   19 Oct  7 23:20 dirl
-rwxr-xr-x. 1 guest guest 25952 Oct  8 00:12 readfile
-rw-----. 1 root  root   422 Oct  8 00:12 readfile.c
-rwxr-xr-x. 1 guest guest 25904 Oct  7 23:50 simpleid
-rwsr-sr-x. 1 root  guest 26008 Oct  7 23:56 simpleid2
-rw-r--r--. 1 guest guest   313 Oct  7 23:56 simpleid2.c
-rw-r--r--. 1 guest guest   181 Oct  7 23:50 simpleid.c
[root@doevseeva guest]#
[root@doevseeva guest]# exit
exit
[guest@doevseeva ~]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@doevseeva ~]$
```

Рис. 13: Смена владельца файла и изменение прав

Далее сменим у программы readfile владельца и установим SetUID-бит.

```
[guest@doevseeva ~]$ su
Password:
[root@doevseeva guest]# chown root /home/guest/readfile
[root@doevseeva guest]# chmod u+s /home/guest/readfile
[root@doevseeva guest]#
```

Рис. 14: Смена владельца и установка SetUID-бита

Проверим возможность чтения программой readfile файлов readfile.c и /etc/shadow от имени суперпользователя и пользователя guest.

```
[root@doevseeva guest]#
[root@doevseeva guest]# ./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[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. 15: Чтение readfile.c от имени суперпользователя

```
[root@doevseeva guest]# ./readfile /etc/shadow
root:$6$WBTn0LvqyZ30L6dS$vtWYhTU0Y9ABTAVak9ZDgcwLHWG03eq4/w15KVUZVj5m9Lx9PDFGeb
D.n8UuzEO58z9l53lMvm4dh6C9QihF/:0:99999:7:::
bin:!:19123:0:99999:7:::
daemon:!:19123:0:99999:7:::
adm:!:19123:0:99999:7:::
lp:!:19123:0:99999:7:::
sync:!:19123:0:99999:7:::
shutdown:!:19123:0:99999:7:::
halt:!:19123:0:99999:7:::
mail:!:19123:0:99999:7:::
operator:!:19123:0:99999:7:::
games:!:19123:0:99999:7:::
ftp:!:19123:0:99999:7:::
nobody:!:19123:0:99999:7:::
systemd-coredump:!!:19245:!!!!:
dbus:!!:19245:!!!!:
polkitd:!!:19245:!!!!:
rtkit:!!:19245:!!!!:
sssd:!!:19245:!!!!:
avahi:!!:19245:!!!!:
pipewire:!!:19245:!!!!:
libstoragemgmt:!!:19245:!!!!:
```

Рис. 16: Чтение /etc/shadow от имени суперпользователя

```
[root@doevseeva guest]# exit
exit
[guest@doevseeva ~]$ ./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[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. 17: Чтение readfile.c от имени пользователя guest

```
[guest@doevseeva ~]$ ./readfile /etc/shadow
root:$6$WBTn0LvqyZ30L6dS$svQtWyhTU0Y9ABTAVak9ZDgcwlHWG03eq4/w15KVUZVj5m9Lx9PDFGeb
D.n8UuzE0S8z9l53lMvm4dh6C9QihF/:0:99999:7:::
bin:!:19123:0:99999:7:::
daemon:!:19123:0:99999:7:::
adm:!:19123:0:99999:7:::
lp:!:19123:0:99999:7:::
sync:!:19123:0:99999:7:::
shutdown:!:19123:0:99999:7:::
halt:!:19123:0:99999:7:::
mail:!:19123:0:99999:7:::
operator:!:19123:0:99999:7:::
games:!:19123:0:99999:7:::
ftp:!:19123:0:99999:7:::
nobody:!:19123:0:99999:7:::
systemd-coredump:!:19245::::::
dbus:!:19245::::::
polkitd:!:19245::::::
rtkit:!:19245::::::
sssd:!:19245::::::
avahi:!:19245::::::
pipewire:!:19245::::::
libstoragemgmt:!:19245::::::
```

Рис. 18: Чтение /etc/shadow от имени пользователя guest

Как мы видим, файлы были успешно прочитаны в обоих случаях.

## 2. Работа со Sticky-битом

Выясним, установлен ли атрибут Sticky на директории /tmp.

```
[guest@doevseeva ~]$
[guest@doevseeva ~]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 Oct 8 00:18 tmp
[guest@doevseeva ~]$
```

Рис. 19: Проверка наличия атрибута Sticky

Мы видим, что атрибут t установлен на директории.

От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test, посмотрим его атрибуты и разрешим чтение и запись для категории пользователей 'все остальные'.

```
[guest@doevseeva ~]$
[guest@doevseeva ~]$ echo "test" > /tmp/file01.txt
[guest@doevseeva ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Oct 8 00:26 /tmp/file01.txt
[guest@doevseeva ~]$ chmod o+rw /tmp/file01.txt
[guest@doevseeva ~]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct 8 00:26 /tmp/file01.txt
[guest@doevseeva ~]$
```

Рис. 20: Создание файла и изменение прав

От имени пользователя guest2 попробуем прочитать созданный файл.

```
[guest2@doevseeva ~]$ su - guest2
Password:
[guest2@doevseeva ~]$
[guest2@doevseeva ~]$ cat /tmp/file01.txt
test
[guest2@doevseeva ~]$
```

Рис. 21: Чтение файла от имени пользователя guest2

Файл был успешно прочитан.

Далее попробуем сделать дозапись в файл, перезаписать его содержимое и удалить файл.

```
[guest2@doevseeva ~]$ echo "test2" >> /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
[guest2@doevseeva ~]$ cat /tmp/file01.txt
test
[guest2@doevseeva ~]$ █
```

Рис. 22: Попытка дозаписи в файл

```
[guest2@doevseeva ~]$ echo "test3" > /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
[guest2@doevseeva ~]$ cat /tmp/file01.txt
test
[guest2@doevseeva ~]$
```

Рис. 23: Попытка перезаписи файла

```
[guest2@doevseeva ~]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@doevseeva ~]$ █
```

Рис. 24: Попытка удаления файла

При попытке выполнения каждой из команд было отказано в доступе, соответственно выполнить их не удалось.

Повысим свои права до суперпользователя и снимем атрибут t с директории /tmp.

```
[guest2@doevseeva ~]$ su -
Password:
[root@doevseeva ~]# chmod -t /tmp
[root@doevseeva ~]# █
```

Рис. 25: Снятие атрибута Sticky

Покинем режим суперпользователя и проверим успешность снятия атрибута.

```
[root@doevseeva ~]# exit
logout
[guest2@doevseeva ~]$ ls -l / | grep tmp
drwxrwxrwx. 15 root root 4096 Oct 8 00:30 tmp
[guest2@doevseeva ~]$
```

Рис. 26: Проверка атрибутов

Далее попробуем выполнить команды, которые не удалось выполнить ранее.

```
[guest2@doevseeva ~]$ cat /tmp/file01.txt
test
[guest2@doevseeva ~]$ echo "test2" >> /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
[guest2@doevseeva ~]$ cat /tmp/file01.txt
test
[guest2@doevseeva ~]$ echo "test3" > /tmp/file01.txt
-bash: /tmp/file01.txt: Permission denied
[guest2@doevseeva ~]$ cat /tmp/file01.txt
test
[guest2@doevseeva ~]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
[guest2@doevseeva ~]$
```

Рис. 27: Попытка повторного выполнения команд

Итак, мы можем видеть, что при попытке дозаписи в файл или перезаписи его содержимого в доступе снова было отказано, однако на этот раз удалось успешно удалить файл от имени пользователя, не являющегося его владельцем.

Повысим свои права до суперпользователя и вернем атрибут `t` на директорию `/tmp`.

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

Рис. 28: Возвращение атрибута Sticky



## Выводы

В результате проделанной работы мы изучили механизмы изменения идентификаторов и применения SetUID-, SetGID- и Sticky-битов, получили практические навыки работы в консоли с дополнительными атрибутами, рассмотрели работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

## Список литературы

- Методические материалы к лабораторной работе, представленные на сайте “ТУИС РУДН” <https://esystem.rudn.ru/>