

Отчёт по лабораторной работе №5

Дисциплина: Информационная безопасность

Евдокимов Иван Андреевич

Содержание

Техническое оснащение:	5
Цель работы:	6
Выполнение лабораторной работы	7
Исследование Sticky-бита	12
Список литературы	16

Список иллюстраций

1	Вошёл в систему	7
2	Программа simpleid.c	7
3	Выполнил системную программу id	7
4	Усложнил программу	8
5	Скомпилировал и запустил simpleid2.c	8
6	Выполнил команды chown root:guest	9
7	SetGID-бит	9
8	Смена владельца и изменение прав программы readfile.c	10
9	Проверка возможности чтения файла readfile.c пользователем guest	11
10	Смена у программы readfile владельца и установка SetU'D-бита .	11
11	Чтение программой readfile файла readfile.c	11
12	Чтение программой readfile файла readfile.c	12
13	Выполнение команды ls -l / grep tmp и создание файла file01.txt .	12
14	Арибуты файла file01.txt	13
15	Чтение, дозапись, запись, удаление файл /tmp/file01.txt	13
16	Сняли Sticky-бит с директории /tmp	13
17	Отсутствие атрибута t у директории /tmp	14
18	Чтение, дозапись, запись, удаление файл /tmp/file01.txt без атрибута t	14
19	Чтение, дозапись, запись, удаление файл /tmp/file01.txt без атрибута t	14

Список таблиц

Техническое оснащение:

- Персональный компьютер с операционной системой Windows 10;
- OBS Studio, использующийся для записи скринкаста лабораторной работы;
- Приложение Visual Studio Code для редактирования файлов формата *md*, а также для конвертации файлов отчётов и презентаций;

Цель работы:

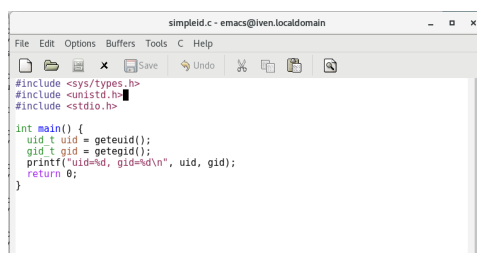
Целью данной лабораторной работы является изучить механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Выполнение лабораторной работы

1. Вошёл в систему от имени пользователя guest и создал программу simpleid.c.

```
[eangreevich@iven ~]$ su guest
Пароль:
[guest@iven eangreevich]$ touch simpleid.c
```

Рис. 1: Вошёл в систему



```
simpleid.c - emacs@iven.localdomain
File Edit Options Buffers Tools C Help
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int main() {
    uid_t uid = getuid();
    gid_t gid = getgid();
    printf("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

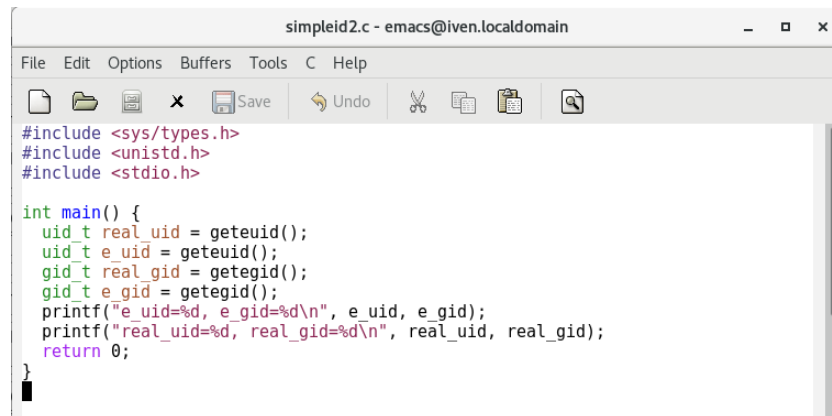
Рис. 2: Программа simpleid.c

2. Скомпилировал программу и убедился, что файл программы создан. Выполнял программу simpleid. Выполнял системную программу id. В отличие от команды id, моя программа не выводит контекст и все группы, в которые пользователь.

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

Рис. 3: Выполнил системную программу id

3. Усложнил программу, добавив вывод действительных идентификаторов.



```
simpleid2.c - emacs@iven.localdomain
File Edit Options Buffers Tools C Help
[Icons: Save, Undo, Cut, Copy, Paste, Find]

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

int main() {
    uid_t real_uid = geteuid();
    uid_t e_uid = geteuid();
    gid_t real_gid = getegid();
    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. 4: Усложнил программу

4. Получившуюся программу назвал simpleid2.c. Скомпилировал и запустил simpleid2.c.

Скомпилировал и запустил simpleid2.c

Рис. 5: Скомпилировал и запустил simpleid2.c

5. От имени суперпользователя выполнил команды `chown root:guest /home/guest/simpleid2` и `chmod u+s /home/guest/simpleid2`. Первая команда меняет владельца файла simpleid2 на группу guest. Вторая команда меняет права доступа к файлу simpleid2 для пользователя и установленные атрибуты SUID или SGID позволяют запускать файл на выполнение с правами владельца файла или группы соответственно. Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2. Запустил simpleid2 и `id`. Сравнил результаты.

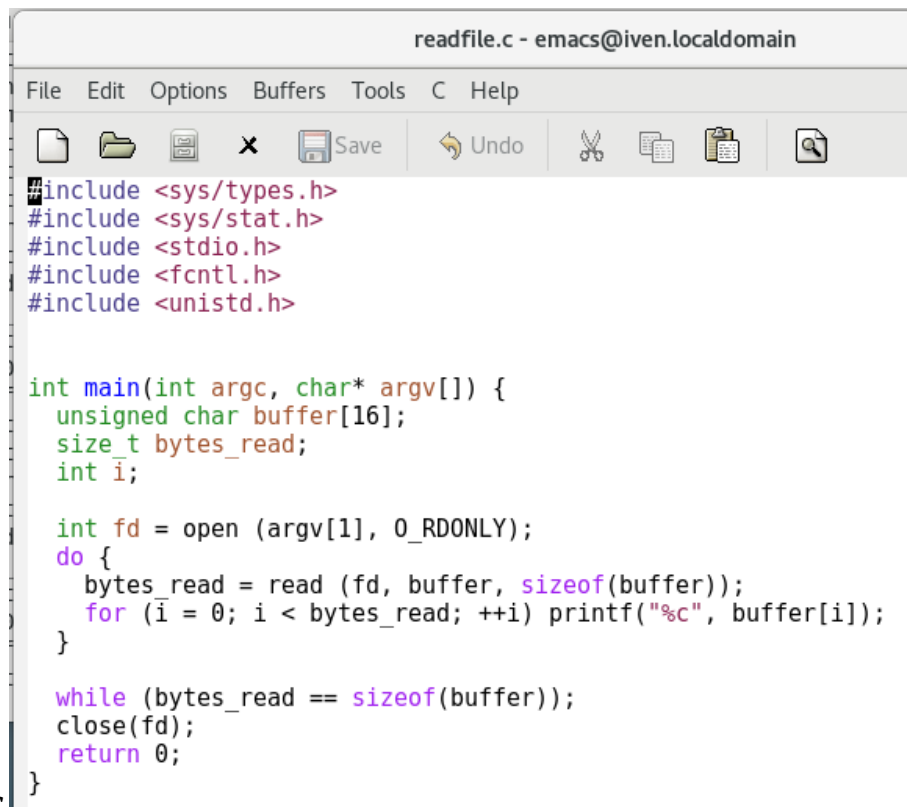

```
[root@iven guest]# chown root:guest /home/guest/simpleid2
[root@iven guest]# chmod u+s /home/guest/simpleid2
[root@iven guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8512 окт 7 17:41 simpleid2
[root@iven guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@iven guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@iven guest]#
```

Рис. 6: Выполнил команды chown root:guest

6. Прodelал тоже самое относительно SetGID-бита. Значения вывода обеих программ совпадают, только в отличие от предыдущего пункта значение `e_gid = 1002. (@fig:007)`

```
[root@iven guest]# chmod g+s /home/guest/simpleid2
[root@iven guest]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 8512 окт 7 17:41 simpleid2
[root@iven guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=1001
[root@iven guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@iven guest]#
```

Рис. 7: SetGID-бит



```
#include <sys/types.h>
#include <sys/stat.h>
#include <stdio.h>
#include <fcntl.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;
}
```

7. Создал программу readfile.c

8. Откомпилировали программу readfile.c: **gcc readfile.c -o readfile**. Сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. (@fig:009)

```
[guest@iven ~]$ su root
Пароль:
[root@iven guest]# chowd root /home/guest/readfile.c
bash: chowd: команда не найдена...
[root@iven guest]# chown root /home/guest/readfile.c
[root@iven guest]# chmod 700 /home/guest/readfile.c
[root@iven guest]#
```

Рис. 8: Смена владельца и изменение прав программы readfile.c

10. Проверили, что пользователь guest не может прочитать файл readfile.c. (@fig:010)

```
[root@iven ~]# su guest
[guest@iven ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@iven ~]$
```

Рис. 9: Проверка возможности чтения файла readfile.c пользователем guest

11. Сменили у программы readfile владельца и установили SetU'D-бит.
(@fig:011)

```
[guest@iven ~]$ su root
Пароль:
[root@iven ~]# chown root:guest /home/guest/readfile.c
[root@iven ~]# chmod u+s /home/guest/readfile.c
[root@iven ~]#
```

Рис. 10: Смена у программы readfile владельца и установка SetU'D-бита

12. Проверяю, может ли программа readfile прочитать файл readfile.c. Да, может.
(@fig:012)

```
[root@iven ~]# ./readfile readfile.c
#include <sys/types.h>
#include <sys/stat.h>
#include <stdio.h>
#include <fcntl.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. 11: Чтение программой readfile файла readfile.c

13. Проверяю, может ли программа readfile прочитать файл /etc/shadow. Да, может. (@fig:013)

```
[root@iven guest]# ./readfile /etc/shadow
root:$6$C39zanXbwzoBKddR$971dItyQK81TN43uGiV29iKyZoy/yPjiSnDa1a.v9HdpXjVQJ4ujGCJ
ZMX4r56QZo0C1.5s.uzxpqCEGAqn0E/::0:99999:7:::
bin:!:18353:0:99999:7:::
daemon:!:18353:0:99999:7:::
adm:!:18353:0:99999:7:::
lp:!:18353:0:99999:7:::
sync:!:18353:0:99999:7:::
shutdown:!:18353:0:99999:7:::
halt:!:18353:0:99999:7:::
mail:!:18353:0:99999:7:::
operator:!:18353:0:99999:7:::
games:!:18353:0:99999:7:::
ftp:!:18353:0:99999:7:::
nobody:!:18353:0:99999:7:::
systemd-network:!!:19607:.....:
dbus:!!:19607:.....:
polkitd:!!:19607:.....:
libstoragemgmt:!!:19607:.....:
colord:!!:19607:.....:
rpc:!!:19607:0:99999:7:::
saned:!!:19607:.....:
saslauthd:!!:19607:.....:
```

Рис. 12: Чтение программой readfile файла readfile.c

Исследование Sticky-бита

1. Выяснили, что установлен атрибут Sticky на директории /tmp с помощью команды **ls -l / | grep tmp**. От имени пользователя guest создали файл file01.txt в директории /tmp со словом test. (@fig:014)

```
[root@iven guest]# ls -l |grep tmp
[root@iven guest]# su guest
[guest@iven ~]$ echo "test" > /tmp/file01.txt
[guest@iven ~]$ █
```

Рис. 13: Выполнение команды ls -l / | grep tmp и создание файла file01.txt

2. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные». (@fig:015)

```
[guest@iven ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт  7 18:20 /tmp/file01.txt
[guest@iven ~]$ chmod o+rw /tmp/file01.txt
[guest@iven ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 окт  7 18:20 /tmp/file01.txt
[guest@iven ~]$ █
```

Рис. 14: Атрибуты файла file01.txt

- От пользователя guest2 (не являющегося владельцем) попробовали прочитать, дозаписать, записать, удалить файл /tmp/file01.txt. Выполнено все, кроме удаления файла. (@fig:016)

```
[guest@iven ~]$ su guest2
Пароль:
[guest2@iven guest]$ cat /tmp/file01.txt
test
[guest2@iven guest]$ echo "test2" >> /tmp/file01.txt
[guest2@iven guest]$ cat /tmp/file01.txt
test
test2
[guest2@iven guest]$ echo "test3" > /tmp/file01.txt
[guest2@iven guest]$ cat /tmp/file01.txt
test3
[guest2@iven guest]$ rm /tmp/file01.txt
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена
[guest2@iven guest]$ █
```

Рис. 15: Чтение, дозапись, запись, удаление файл /tmp/file01.txt

- Повысили свои права до суперпользователя командой **su** - и выполнили после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: **chmod -t /tmp**. Покинули режим суперпользователя командой **exit**.(@fig:017)

```
[guest2@iven guest]$ su -
Пароль:
Последний вход в систему:Сб окт  7 18:15:31 MSK 2023на pts/2
[root@iven ~]# chmod -t /tmp
[root@iven ~]# exit
logout
█
```

Рис. 16: Сняли Sticky-бит с директории /tmp

5. От пользователя guest2 проверили, что атрибута t у директории /tmp нет. (@fig:018)

```
[guest2@iven guest]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 окт  7 18:27 tmp
[guest2@iven guest]$
```

Рис. 17: Отсутствие атрибута t у директории /tmp

6. Повторили предыдущие шаги. В данном случае получилось выполнить удаление файла. (@fig:019)

```
[guest2@iven guest]$ cat /tmp/file01.txt
test3
[guest2@iven guest]$ echo "test2" >> /tmp/file01.txt
[guest2@iven guest]$ cat /tmp/file01.txt
test3
test2
[guest2@iven guest]$ echo "test3" > /tmp/file01.txt
[guest2@iven guest]$ cat /tmp/file01.txt
test3
[guest2@iven guest]$ rm /tmp/file01.txt
[guest2@iven guest]$
```

Рис. 18: Чтение, дозапись, запись, удаление файл /tmp/file01.txt без атрибута t

7. Повысили свои права до суперпользователя и вернули атрибут t на директорию /tmp. (@fig:020)

```
[root@iven guest]# chmod g+s /home/guest/simpleid2
[root@iven guest]# ls -l simpleid2
-rwsrwsr-x. 1 root guest 8512 окт  7 17:41 simpleid2
[root@iven guest]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=1001
[root@iven guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@iven guest]#
```

Рис. 19: Чтение, дозапись, запись, удаление файл /tmp/file01.txt без атрибута t

Выводы:

В ходе выполнения лабораторной работы мы приобрели изучили механизмы изменения идентификаторов, применения SetUID- 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

1. Официальный сайт VirtualBox
2. Материал для выполнения лабораторной
3. Официальный сайт CentOS