

Лабораторная работа № 5

Дисциплина: Информационная безопасность

Новосельцев Данила Сергеевич

Содержание

1 Цель работы	4
2 Выполнение лабораторной работы	5
2.1 Создание программы	5
2.2 Исследование Sticky-бита	10
3 Вывод	13

Список иллюстраций

2.1	Код программы в редакторе Emacs	5
2.2	Успешная компиляция и запуск simpleid. Запуск системной программы id	6
2.3	Код программы в редакторе Emacs	6
2.4	Успешная компиляция	6
2.5	Успешная компиляция с новым владельцем файла	7
2.6	Код программы в редакторе Emacs	8
2.7	Смена прав и владельца и проверка чтения файла	8
2.8	Смена прав и владельца	9
2.9	Проверка функционала файла	9
2.10	Чтение файла /etc/shadow	10
2.11	Смена прав и владельца	10
2.12	проверка атрибутов программы	10
2.13	Редактирование файла	11
2.14	Редактирование файла в режиме суперпользователя	12

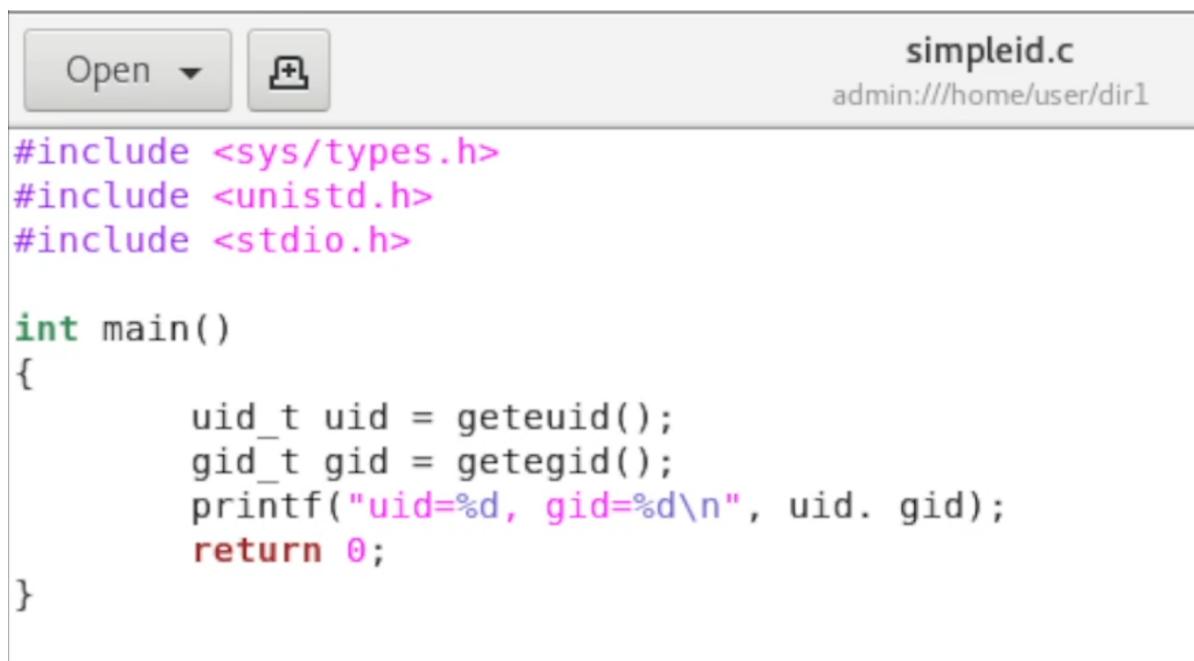
1 Цель работы

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 Выполнение лабораторной работы

2.1 Создание программы

1. Вошёл в систему от имени пользователя guest и создал программу simpleid.c(2.1).



The screenshot shows a window of the Emacs text editor. The title bar reads "simpleid.c" and "admin:///home/user/dirl". On the left, there are two buttons: "Open" with a dropdown arrow and a small icon. The main area contains the C program code:

```
#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;
}
```

Рис. 2.1: Код программы в редакторе Emacs

2. Скомпилировал и выполнил программу simpleid.(2.2).

```
[root@user dir1]# gcc simpleid.c -o simpleid
[root@user dir1]# ./simpleid
uid=0, gid=0
```

Рис. 2.2: Успешная компиляция и запуск simpleid. Запуск системной программы id

3. Усложнил программу, добавив вывод действительных идентификаторов(2.3).

```
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;
}
```

Рис. 2.3: Код программы в редакторе Emacs

4. Скомпилировал и запустил simpleid2.c(2.4).

```
[root@user dir1]# gcc simpleid.c -o simpleid
[root@user dir1]# ./simpleid
e_uid=0, e_gid=0
real_uid=0, real_gid=0
```

Рис. 2.4: Успешная компиляция

5. От имени суперпользователя выполнил команды:

```
chown root:guest /home/guest/simpleid2
chmod u+s /home/guest/simpleid2
```

Далее я выполнил проверку правильности установки новых атрибутов и смены владельца файла simpleid2, после чего запустил simpleid2 и id. И проделал то же самое относительно SetGID-бита(2.5).

```
[root@user dir1]# chown root:guest simpleid
[root@user dir1]# chmod u+s simpleid
bash: chmod: command not found...
Similar command is: 'kmod'
[root@user dir1]# chmod u+s simpleid
[root@user dir1]# ls -l simpleid
-rwsr-xr-x. 1 root guest 8616 Nov 12 01:34 simpleid
[root@user dir1]# ./simpleid
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@user dir1]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:u
0-s0:c0.c1023
[root@user dir1]# chmod u-s simpleid
[root@user dir1]# chmod g+s simpleid
[root@user dir1]# ls -l simpleid
-rwrxr-sr-x. 1 root guest 8616 Nov 12 01:34 simpleid
[root@user dir1]# ./simpleid
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@user dir1]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:u
0-s0:c0.c1023
```

Рис. 2.5: Успешная компиляция с новым владельцем файла

6. Создал программу readfile.c и успешно скомпилировал её(2.6).

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

int main(int args, 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. 2.6: Код программы в редакторе Emacs

7. Сменил владельца у файла readfile.c и изменил права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог, после чего проверил, что пользователь guest не может прочитать файл readfile.c(2.7).

```
[root@user dir1]# gcc readfile.c -o readfile
[root@user dir1]# chown root readfile.c
[root@user dir1]# chmod 700 readfile.c
[root@user dir1]# si guest
bash: si: command not found...
Similar command is: 'ci'
[root@user dir1]# su guest
[guest@user dir1]$ cat readfile.c
cat: readfile.c: Permission denied
```

Рис. 2.7: Смена прав и владельца и проверка чтения файла

8. Сменил у программы readfile владельца и установил SetU'D-бит, после чего проверил, может ли программа readfile прочитать файлы readfile.c и /etc/shadow(2.8 - 2.9 - 2.10).

```
[root@user dir1]# chown root:guest readfile
[root@user dir1]# chmod u+s readfile
[root@user dir1]# su guest
```

Рис. 2.8: Смена прав и владельца

```
[root@user dir1]# ./readfile readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

int main(int args, 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. 2.9: Проверка функционала файла

Рис. 2.10: Чтение файла /etc/shadow

2.2 Исследование Sticky-бита

- Выяснил, установлен ли атрибут Sticky на директории /tmp. От имени пользователя guest я создал файл file01.txt в директории /tmp со словом test, далее просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей «все остальные»(2.11 - 2.12).

```
[guest@user dir1]$ ls -l / | grep tmp  
drwxrwxrwt. 16 root root 4096 Nov 12 08:47 tmp  
[guest@user dir1]$ echo "test" >> /tmp/file01.txt  
[guest@user dir1]$ chmod o+rw /tmp/file01.txt  
chmod: cannot access '/tmp/file01.txt': No such file or direc  
[guest@user dir1]$ chmod o+rw /tmp/file01.txt
```

Рис. 2.11: Смена прав и владельца

```
[guest@user dir1]$ ls -l /tmp/file01.txt  
-rw-r--rw-. 1 root root 10 Nov 12 08:52 /tmp/file01.txt  
[guest@user dir1]$ █
```

Рис. 2.12: проверка атрибутов программы

2. От пользователя guest2 я прочитал файл /tmp/file01.txt и дозаписал в файл слово test. Далее я записал в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию и проверил содержимое файла командой. Я попробовал удалить файл file01.txt(2.13).

```
[guest@user dir1]$ su guest2
Password:
[guest2@user dir1]$ cat tmp/file01.txt
cat: tmp/file01.txt: No such file or directory
[guest2@user dir1]$ cat /tmp/file01.txt
test
test
[guest2@user dir1]$ echo "test" /tmp/file01.txt
test /tmp/file01.txt
[guest2@user dir1]$ echo "test" >> /tmp/file01.txt
[guest2@user dir1]$ cat /tmp/file01.txt
test
test
test
[guest2@user dir1]$ echo "test3" > /tmp/file01.txt
[guest2@user dir1]$ cat /tmp/file01.txt
test3
[guest2@user dir1]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
```

Рис. 2.13: Редактирование файла

Однако в ответ я получил отказ от выполнения операции.

3. Повысил свои права до суперпользователя, затем снял атрибут t с директории /tmp, после чего покинул режим суперпользователя командой exit. От имени пользователя guest2 проверил, что атрибута t у директории /tmp нет, после чего повторил те же действия. Теперь я могу удалить файл file01.txt. Повысил свои права до суперпользователя и вернул атрибут t на директорию /tmp(2.14).

```
[guest2@user dir1]$ su
Password:
[root@user dir1]# chmod -t /tmp
[root@user dir1]# exit
exit
[guest2@user dir1]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 Nov 12 09:02 tmp
[guest2@user dir1]$ rm /tmp/file01.txt
[guest2@user dir1]$ su
Password:
[root@user dir1]# chmod +t /tmp
[root@user dir1]# exit
exit
[guest2@user dir1]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Nov 12 09:03 tmp
```

Рис. 2.14: Редактирование файла в режиме суперпользователя

3 Вывод

Изучил механизмы изменения идентификаторов, применения SetUID- и Sticky-битов, получил практических навыков работы в консоли с дополнительными атрибутами, а также рассмотрел работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.