
Информационная безопасность

Лабораторная работа №5

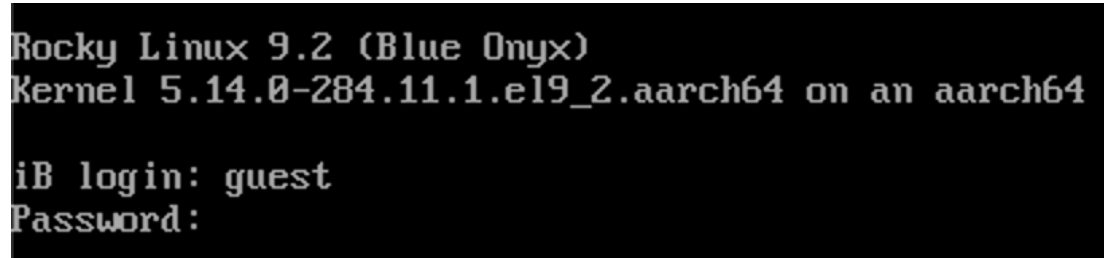
Кузнецов Юрий Владимирович

Цель работы:

Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в кон- соли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Ход работы:

1. Войдите в систему от имени пользователя guest.



```
Rocky Linux 9.2 (Blue Onyx)
Kernel 5.14.0-284.11.1.el9_2.aarch64 on an aarch64

iB login: guest
Password:
```

рис.1

2. Создайте программу 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;
}
```

рис.2

3. Скомпилируйте программу и убедитесь, что файл программы создан: gcc simpleid.c -o simpleid

```
[guest@iB Lab05]$ ./simpleid
uid=1000, gid=1000
[guest@iB Lab05]$
```

рис.3

4. Выполните программу simpleid:

```
[guest@iB Lab05]$ ./simpleid
uid=1000, gid=1000
[guest@iB Lab05]$ id
uid=1000(guest) gid=1000(guest) groups=1000(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@iB Lab05]$ _
```

рис.4

5. Выполните системную программу id: id и сравните полученный вами результат с данными предыдущего пункта задания.

```
#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 = getegid ();
    gid_t e_gid = getgid ();

    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.5

6. Усложните программу, добавив вывод действительных идентификаторов:

```
[guest@iB Lab05]$ ./simpleid2
e_uid=0, e_gid=1000
real_uid=1000, real_gid=0
[guest@iB Lab05]$ _
```

рис.6

7. Скомпилируйте и запустите simpleid2.c: gcc simpleid2.c -o simpleid2 ./simpleid2
8. От имени суперпользователя выполните команды: chown root:guest /home/guest/simpleid2 chmod u+s /home/guest/simpleid2
9. Используйте sudo или повысьте временно свои права с помощью su. Поясните, что делают эти команды.

10. Выполните проверку правильности установки новых атрибутов имени владельца файла simpleid2: `ls -l simpleid2`
11. Запустите simpleid2 и id: `./simpleid2 id`
12. Прodelайте тоже самое относительно SetGID-бита.
13. Создайте программу readfile.c:
14. Откомпилируйте её: `gcc readfile.c -o readfile`
15. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.
16. Проверьте, что пользователь guest не может прочитать файл readfile.c.
17. Смените у программы readfile владельца и установите SetU'D-бит.
18. Проверьте, может ли программа readfile прочитать файл readfile.c?
19. Проверьте, может ли программа readfile прочитать файл /etc/shadow? Отразите полученный результат и ваши объяснения в отчёте.
20. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду `ls -l / | grep tmp`
21. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test: `echo "test" > /tmp/file01.txt`
22. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»: `ls -l /tmp/file01.txt chmod o+rw /tmp/file01.txt ls -l /tmp/file01.txt`
23. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt: `cat /tmp/file01.txt`
24. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой `echo "test2" > /tmp/file01.txt` Удалось ли вам выполнить операцию?
25. Проверьте содержимое файла командой `cat /tmp/file01.txt`
26. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой `echo "test3" > /tmp/file01.txt` Удалось ли вам выполнить операцию?
27. Проверьте содержимое файла командой `cat /tmp/file01.txt`
28. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt` Удалось ли вам удалить файл?

29. Повысьте свои права до суперпользователя следующей командой `su -` и выполните после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`: `chmod -t /tmp`
30. Покиньте режим суперпользователя командой `exit`
31. От пользователя `guest2` проверьте, что атрибута `t` у директории `/tmp` нет: `ls -l / | grep tmp`
32. Повторите предыдущие шаги. Какие наблюдаются изменения?
33. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт.
34. Повысьте свои права до суперпользователя и верните атрибут `t` на директорию `/tmp`: `su - chmod +t /tmp exit`

Вывод:

Мы изучили 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.