Отчёт по лабораторной работе 5

Макухина Анастасия Вадимовна

Содержание

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.

Выполнение лабораторной работы

1. Войдём в систему от имени пользователя guest

Рисунок 1.

2. Создайте программу simpleid.c:

nano 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”, uid, gid);
return 0;
}
```

Рисунок 2, Рисунок 3.

3. Скомплилируем программу и убедимся, что файл программы создан:

gcc simpleid.c -o simpleid - Рисунок 4.

4. Выполним программу simpleid:

./simpleid - Рисунок 5.

5. Выполните системную программу id:

id - Рисунок 6.

Результаты одинаковы.

6. Усложним программу, добавив вывод действительных идентификаторов:

```
#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”, e_uid, e_gid);
printf (“real_uid=%d, real_gid=%d”, real_uid,
,→ real_gid);
return 0;
}
```

Получившуюся программу назовём simpleid2.c - Рисунок 7.

7. Скомпилируем и запустим simpleid2.c:

gcc simpleid2.c -o simpleid2

./simpleid2 - Рисунок 8.

8. От имени суперпользователя выполним команды:

chown root:guest /home/guest/simpleid2

chmod u+s /home/guest/simpleid2 - Рисунок 9.

9. Повысим временно свои права с помощью su.

su - Рисунок 10.

10. Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

ls -l simpleid2 - Рисунок 11.

11. Запустим simpleid2 и id:

./simpleid2

id - Рисунок 12.

Результаты одинаковы.

12. Проделайте тоже самое относительно SetGID-бита - Рисунок 13.

13. Создадим 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;

}

Рисунок 14.

14. Откомпилируем её

gcc readfile.c -o readfile - Рисунок 15.

15. Сменим владельца у файла readfile.c (или любого другого текстового файла в системе) и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.

Рисунок 16.

16. Проверим, что пользователь guest не может прочитать файл readfile.c

Рисунок 17.

17. Сменим у программы readfile владельца и установим SetU'D-бит

Рисунок 18.

18. Проверим, может ли программа readfile прочитать файл readfile.c

Рисунок 19, Рисунок 20.

19. Проверим, может ли программа readfile прочитать файл /etc/shadow

Рисунок 21.

20. Выясним, установлен ли атрибут Sticky на директории /tmp, для чего выполним команду:

ls -l / | grep tmp - Рисунок 22.

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

echo "test" > /tmp/file01.txt - Рисунок 23.

22. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные»:

ls -l /tmp/file01.txt

chmod o+rw /tmp/file01.txt

ls -l /tmp/file01.txt - Рисунок 24.

23. От пользователя guest2 (не являющегося владельцем) попробуем прочитать файл /tmp/file01.txt:

cat /tmp/file01.txt - Рисунок 25.

24. От пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2 командой:

echo “test2” > /tmp/file01.txt - Рисунок 26.

Да, удалось.

25. Проверим содержимое файла командой:

cat /tmp/file01.txt - Рисунок 27.

26. От пользователя guest2 попробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой:

echo “test3” > /tmp/file01.txt - Рисунок 28.

27. Проверьте содержимое файла командой:

cat /tmp/file01.txt - Рисунок 29.

28. От пользователя guest2 попробуем удалить файл /tmp/file01.txt командой:

rm /tmp/fileOl.txt - Рисунок 30.

Не удалось.

29. Повысим свои права до суперпользователя следующей командой

su

и выполним после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp:

chmod -t /tmp - Рисунок 31.

30. Покинем режим суперпользователя командой

exit - Рисунок 32.

31. От пользователя guest2 проверим, что атрибута t у директории /tmp нет:

ls -l / | grep tmp - Рисунок 33.

32. Повторите предыдущие шаги - Рисунок 34.

Изменений не произошло.

33. Проверим, можно ли удалить файл от имени пользователя, не являющегося его владельцем - Рисунок 35.

Удалось.

34. Повысим свои права до суперпользователя и вернём атрибут t на директорию /tmp:

su -

chmod +t /tmp

exit - Рисунок 36.

Выводы

В ходе выяполнения работы я изученла механизмы изменения идентификаторов, применения SetUID- и Sticky-битов, полученила практические навыки работы в консоли с дополнительными атрибутами, рассмотрела работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.