Презентация по лабораторной работе № 5

Информационная безопасность

Адебайо Р. А.

06.10.2023

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Адебайо Ридвануллахи Айофе
- студент группы НКНбд-01-20
- Факультет физико-математических и естественных наук
- Российский университет дружбы народов
- Страничка на GitHub
- Страничка на LinkedIn

Цель лабораторной работы

Цель лабораторной 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

Создание программы

Сначала созданы и выполнены две программы, имитирующие команду ‘id’, для отображения различных идентификаторов пользователя и группы. Затем, с правами суперпользователя, установлены биты SetUID и SetGID для этих программ. После этого, при выполнении программы, они получают соответствующие привилегии суперпользователя и группы. Это демонстрирует, как изменение битов SetUID и SetGID может повлиять на выполнение программ и их привилегии.

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

Рис. 1: 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;
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"simpleid.c" 11L, 174B
```

Создание программы(2)

Сначала создали программу для чтения файла (readfile.c) и скомпилировали её. Затем изменили права доступа к программе так, чтобы только пользователь root мог её читать, а гость - нет. Убедились, что гость не имеет доступа к файлу readfile.c через выполнение программы. Далее сменили владельца программы readfile и установили бит SetUID. После этого с помощью программы удалось прочитать файлы readfile.c и /etc/shadow. Этот процесс иллюстрирует изменение прав доступа и привилегий программы в системе.

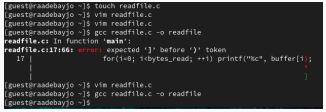


Рис. 3: readfile.c

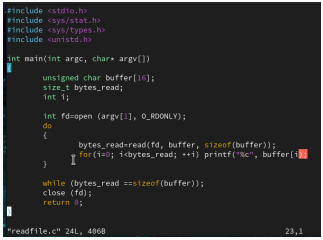


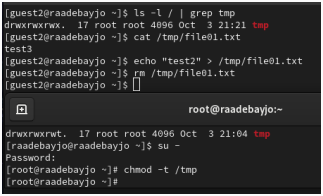
Рис. 4: Код

Исследование Sticky-бита

Сначала мы создали файл в каталоге /tmp, разрешив чтение и запись для всех пользователей. Затем, от имени пользователя guest2, мы попытались прочитать, дозаписать и переписать файл. Однако нам не удалось удалить файл.

Затем, суперпользователь снял Sticky-бит с каталога tmp и мы повторили действия с файлом. В этот раз удаление файла стало возможным.

Наконец, суперпользователь вернул Sticky-бит на каталог tmp, обеспечивая тем самым ограниченный доступ к файлам в этом каталоге, даже для суперпользователя. Эти действия демонстрируют влияние Sticky-бита на возможности удаления файлов в каталоге.



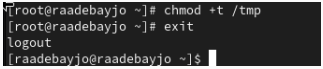


Рис. 6: sticky-bit(2)

Вывод

Вывод

В ходе выполнения данной лабораторной работы я изучил механизмы изменения идентификаторов, применение SetUID- и Sticky-битов. Получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.