Лабораторная работа №5

Дисциплина: основы информационной безопастности

Астраханцева А. А.

Содержание

# 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 Теоретическое введение

**SetUID**

Бит доступа SETUID дает возможность запускать файл на исполнение от имени владельца файла. Например, именно так работает программа sudo. Если мы посмотрим на нее через ls, то увидим что вместо ключа x стоит ключ s. Это и означает, что у данного файла включен или выставлен параметр SETUID. А значит, если обычный пользователь запустит этот файл, то файл будет запущен от имени владельца, то есть root. Важно понимать, что никакого ввода пароля при этом не потребуется. А сам запрос и обработка пароля организованы уже внутри программы sudo [1].

**Sticky bit**

Это разрешение полезно для защиты файлов от случайного удаления в среде, где несколько пользователей имеют права на запись в один и тот же каталог. Если применяется закрепленный sticky bit, пользователь может удалить файл, только если он является пользователем-владельцем файла или каталога, в котором содержится файл. По этой причине он применяется в качестве разрешения по умолчанию для каталога /tmp и может быть полезен также для каталогов общих групп.

Без sticky bit, если пользователь может создавать файлы в каталоге, он также может удалять файлы из этого каталога. В общедоступной групповой среде это может раздражать. Представьте себе пользователей linda и lori, которые оба имеют права на запись в каталог /data/account и получают эти разрешения благодаря участию в группе account. Поэтому linda может удалять файлы, созданные lori, и наоборот.

Когда вы применяете sticky bit, пользователь может удалять файлы, только если выполняется одно из следующих условий:

Пользователь является владельцем файла;  
Пользователь является владельцем каталога, в котором находится файл.

При использовании ls -ld, вы можете видеть sticky bit как t в позиции, где вы обычно видите разрешение на выполнение для других [2].

# 3 Выполнение лабораторной работы

## 3.1 Подготовка лабораторного стенда

Проверяем, установлен ли у нас компилятор qcc командой gcc -v.

Отключаем систему запретов до очередной перезагрузки системы командой setenforce 0. После этого команда getenforce должна выводить Permissive (рис. 1)

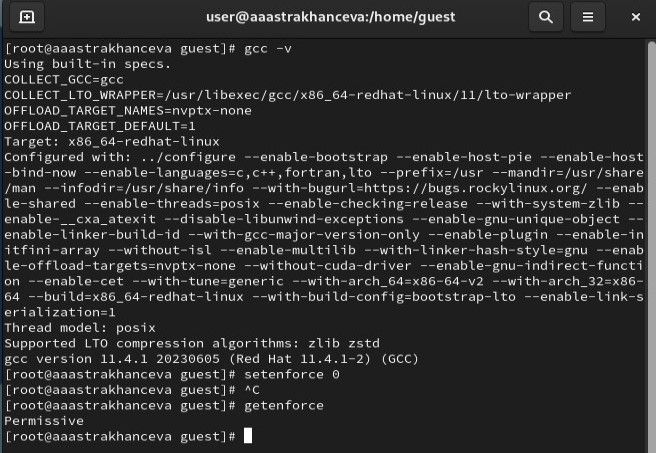


Рис. 1: Подготовка лабораторного стенда

## 3.2 Создание программы

1. Войдите в систему от имени пользователя guest.
2. Создайте программу simpleid.c: (рис. 2)

#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;  
}

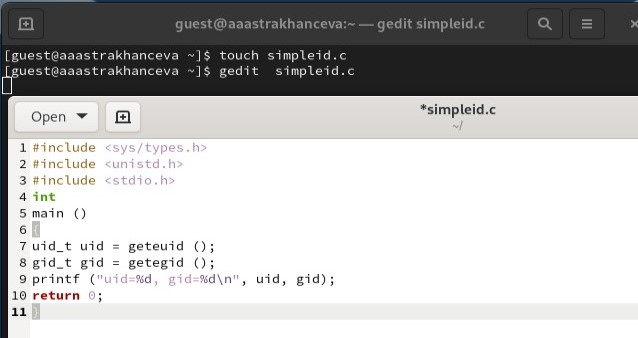


Рис. 2: Создание программы simpleid.c

1. Скомплилируйте программу и убедитесь, что файл программы создан: gcc simpleid.c -o simpleid
2. Выполните программу simpleid: ./simpleid
3. Выполните системную программу id: id и сравните полученный вами результат с данными предыдущего пункта задания. Выводы команд совпали (рис. 3).

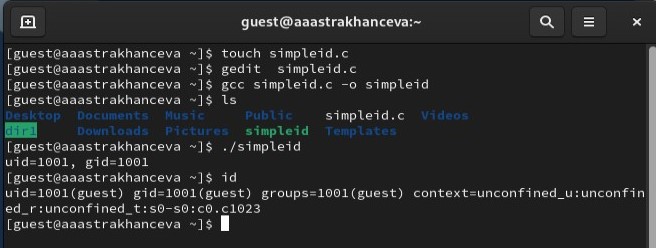


Рис. 3: Компиляция и запуск программы simpleid.c

1. Усложните 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\n", e\_uid, e\_gid);  
printf ("real\_uid=%d, real\_gid=%d\n", real\_uid,  
real\_gid);  
return 0;  
}

Получившуюся программу назовите simpleid2.c. (рис. 4).

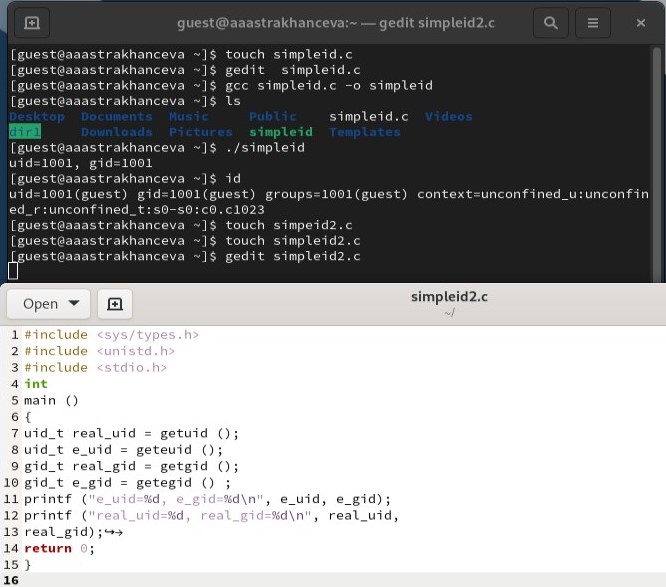


Рис. 4: Создание программы simpleid2.c

1. Скомпилируйте и запустите simpleid2.c: gcc simpleid2.c -o simpleid2, ./simpleid2 (рис. 5).



Рис. 5: Компиляция программы simpleid2.c

1. От имени суперпользователя выполните команды: chown root:guest /home/guest/simpleid2, chmod u+s /home/guest/simpleid2
2. Используйте sudo или повысьте временно свои права с помощью su. Поясните, что делают эти команды. Первая команда меняет владельца файла simpleid2.c, а вторая устанавливает на этот файл SetUID бит.
3. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2: ls -l simpleid2
4. Запустите simpleid2 и id: ./simpleid2, id Сравните результаты. (рис. 6).

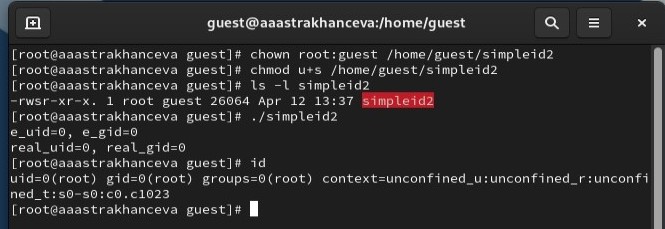


Рис. 6: Изменение владельца и прав доступа для simpleid2

1. Создайте программу readfile.c: (рис. 7).

#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;  
}

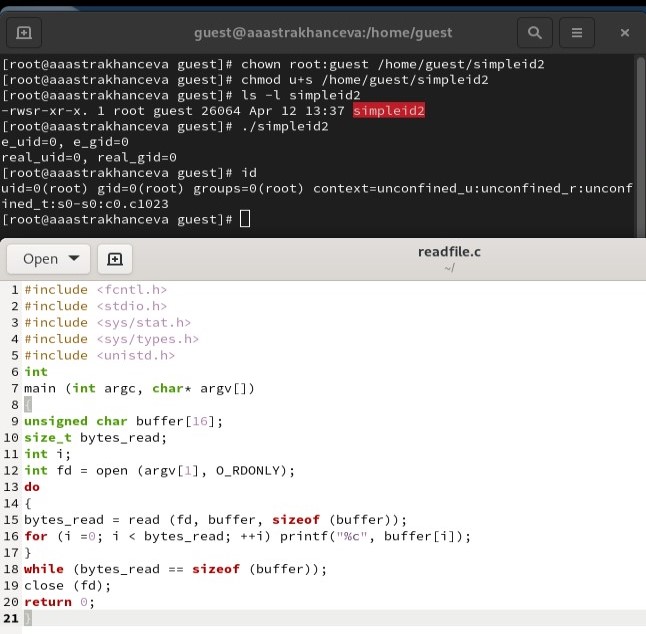


Рис. 7: Создание программы readfile.c

1. Откомпилируйте её. gcc readfile.c -o readfile
2. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог.
3. Проверьте, что пользователь guest не может прочитать файл readfile.c.
4. Смените у программы readfile владельца и установите SetU’D-бит.
5. Проверьте, может ли программа readfile прочитать файл readfile.c?
6. Проверьте, может ли программа readfile прочитать файл /etc/shadow? (рис. 8).

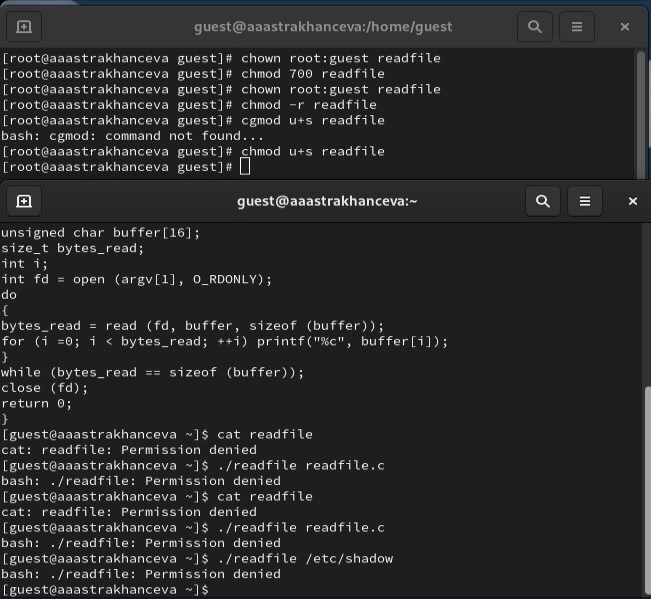


Рис. 8: Пункты 14-19

## 3.3 Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду ls -l / | grep tmp
2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test: echo "test" > /tmp/file01.txt
3. 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 (рис. 9).

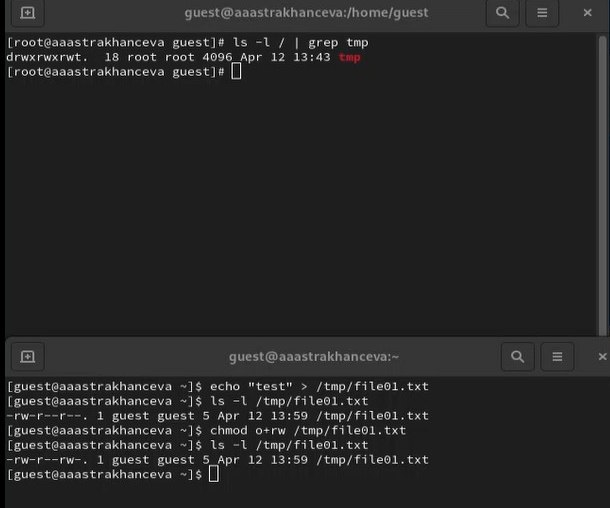


Рис. 9: Проверка установки Stiky бита на дирркетории /tmp, создание нового файла, изменение пра доступа к нему

1. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt: cat /tmp/file01.txt
2. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой echo "test2" > /tmp/file01.txt. Удалось ли вам выполнить операцию?
3. Проверьте содержимое файла командой cat /tmp/file01.txt
4. 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?
5. Проверьте содержимое файла командой cat /tmp/file01.txt
6. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой rm /tmp/fileOl.txt. Удалось ли вам удалить файл? (рис. 10).

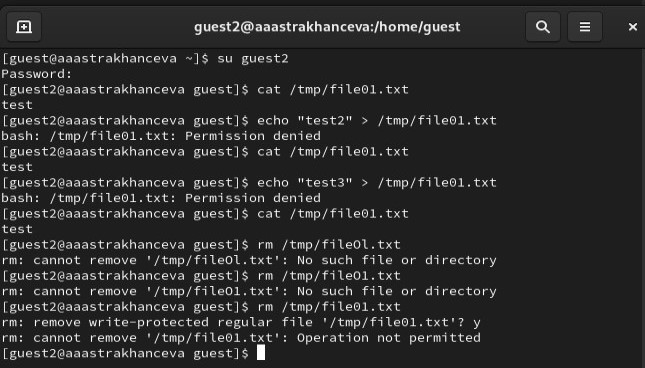


Рис. 10: От лица guest2 попытка прочесть файл, изменить его и удалить его

1. Повысьте свои права до суперпользователя следующей командой su - и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp
2. Покиньте режим суперпользователя командой exit
3. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет: ls -l / | grep tmp (рис. 11).

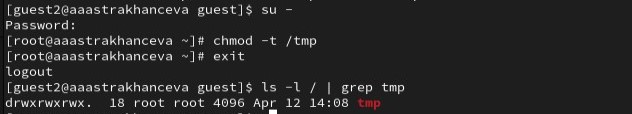


Рис. 11: Переход в режим суепрпользователя, снятие Sticky бита

1. Повторите предыдущие шаги. Какие наблюдаются изменения?
2. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт. Внести изменения в файл не удалось, зато получиось его удалить.
3. Повысьте свои права до суперпользователя и верните атрибут t на директорию /tmp: su - chmod +t /tmp exit (рис. 12).

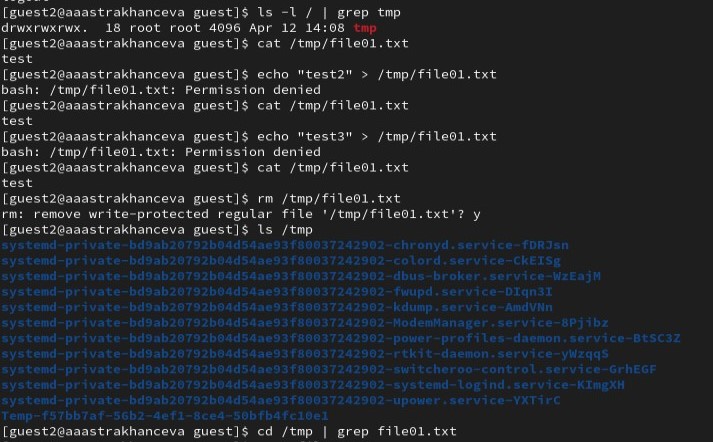


Рис. 12: Повторение пунктов 4-9 со снятым Sticky битом

# 4 Выводы

В ходе выполнения лабораторной рабоыт я изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Также получила 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.

# 5 Список литературы. Библиография

[1] Права доступа на файлы. Часть 3. Специальные биты разрешений SETUID, SETGID и Sticky bit: https://dzen.ru/a/Y-S4apsOlAJ73hJe

[2] Права в Linux (chown, chmod, SUID, GUID, sticky bit, ACL, umask): https://habr.com/ru/articles/469667/