Лабораторная работа № 5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Дисциплина: Информационная безопасность

Манаева Варвара Евгеньевна

Содержание

# 1 Техническое оснащение:

* Персональный компьютер с операционной системой Windows 10;
* Планшет для записи видеосопровождения и голосовых комментариев;
* Microsoft Teams, использующийся для записи скринкаста лабораторной работы;
* Приложение Pycharm для редактирования файлов формата *md*;
* *pandoc* для конвертации файлов отчётов и презентаций.

# 2 Цели и задачи работы

## 2.1 Цель

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.

## 2.2 Задачи

* Настроить по заданию лабораторный стенд;
* Создать программы;
* Подробно разобрать влияние Sticky-бита на удаление и запись файлов.

# 3 Теоретическое введение [1]

Setuid – это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо. Классический пример использования этого бита в операционной системе это команда sudo.

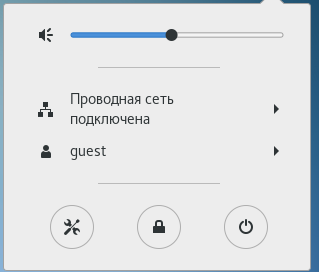
Принцип работы Setgid очень похож на setuid с отличием, что файл будет запускаться пользователем от имени группы, которая владеет файлом.

Последний специальный бит разрешения – это Sticky Bit. В случае, если этот бит установлен для папки, то файлы в этой папке могут быть удалены только их владельцем. Пример использования этого бита в операционной системе это системная папка /tmp. Эта папка разрешена на запись любому пользователю, но удалять файлы в ней могут только пользователи, являющиеся владельцами этих файлов.

# 4 Выполнение лабораторной работы [2]

## 4.1 Создание программ

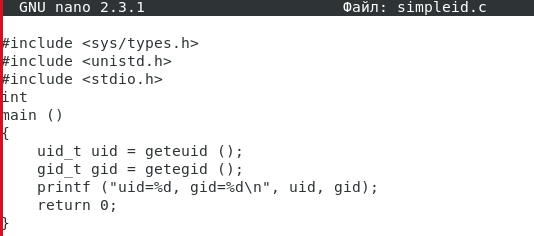
1. Войдём в систему от имени пользователя guest. (??)



Выполнен вход

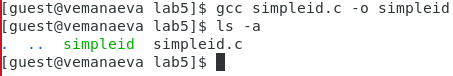
1. Создадим программу simpleid.c (??):

#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;  
}



Создана программа

1. Скомплилируем программу и убедимся, что файл программы создан, командой gcc simpleid.c -o simpleid (??)



Компилируем программу

1. Выполним программу simpleid командой ./simpleid (??)

Вывод программы simpleid

Вывод программы simpleid

1. Выполним системную программу id id (??)

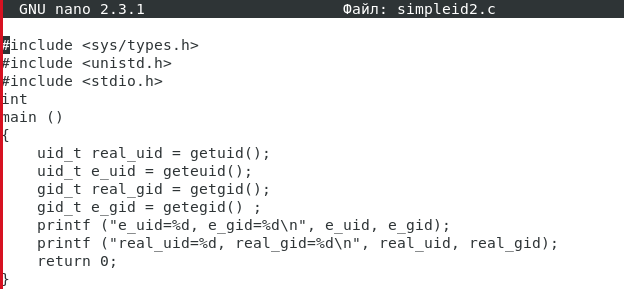
Вывод системной программы

Вывод системной программы

При использовании команды id, в отличие от использования написаной нами программы, выводится не только номер пользователя и номер группы, но и имя пользователя и название группы (в скобках после соответствующих номеров), а также контекст.

1. Усложним программу, добавив вывод действительных идентификаторов, и получившуюся программу назовём simpleid2.c (??).

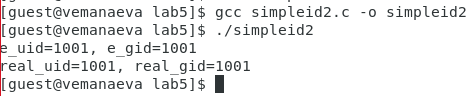
#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
int  
main ()  
{  
 uid\_t real\_uid = getuid();  
 uid\_t e\_uid = geteuid();  
 gid\_t real\_gid = getgid();  
 gid\_t e\_gid = getegid() ;  
 printf ("e\_uid=%d, e\_gid=%d\n", e\_uid, e\_gid);  
 printf ("real\_uid=%d, real\_gid=%d\n", real\_uid, real\_gid);  
 return 0;  
}



Создание второй программы

1. Скомпилируем и запустим simpleid2.c командами (??):

gcc simpleid2.c -o simpleid2  
./simpleid2



Компиляция и запуск второй программы

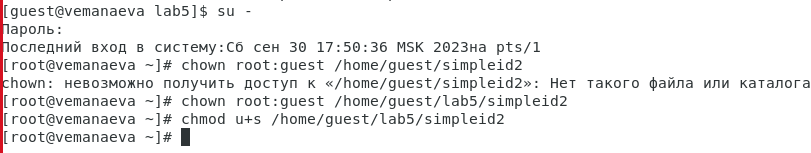
1. Попробуем выполнить команды (??):

chown root:guest /home/guest/lab5/simpleid2  
chmod u+s /home/guest/lab5/simpleid2

Не получается выполнить команды

Не получается выполнить команды

1. Используйте sudo или повысьте временно свои права с помощью su и выполните эти команды. (??)



Меняем пользователя на суперпользователя и выполняем команды

Команда chown root:guest /home/guest/lab5/simpleid2 меняет владельца каталога на root, а группу каталога меняет на группу guest (вернее, оставляет без изменений). Команда chmod u+s /home/guest/lab5/simpleid2 даёт права на выполнение от имени суперпользователя.

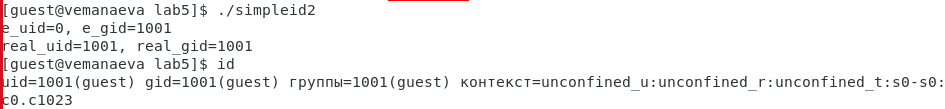
1. Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой ls -l simpleid2 (??)

Проверка

Проверка

1. Запустите simpleid2 и id (??)

id  
./simpleid2

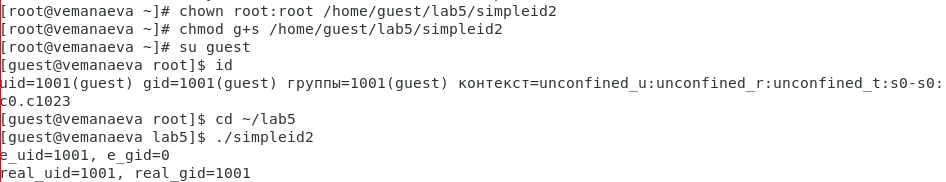


Выполняем программу и системную программу

Команда id, в отличие от написаной программы, выводит только информацию о пользователе, который запросил запуск системной программы, в то время как simpleid2 выводит информацию и о владельце файла, и о том, кто его запускает.

1. Проделайте тоже самое относительно SetGID-бита (??).

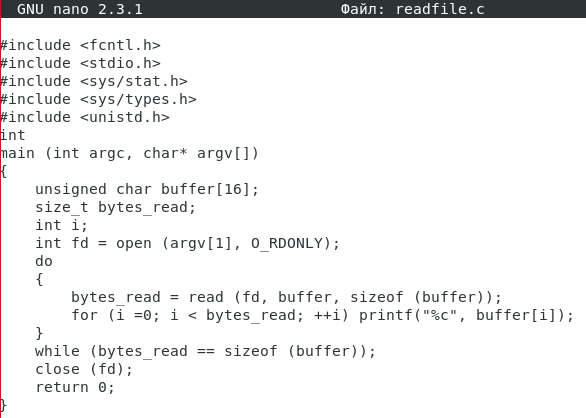
chown root:root /home/guest/lab5/simpleid2  
chmod g+s /home/guest/lab5/simpleid2  
id  
./simpleid2



Повторение действий с SetGID

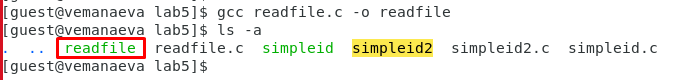
1. Создайте программу readfile.c (??)

#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;  
}



Создаём программу

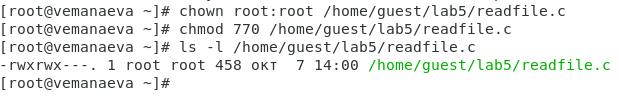
1. Откомпилируйте её командой gcc readfile.c -o readfile (??)



Компилируем программу и проверяем наличие

1. 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 (??).

chown root:root /home/guest/lab5/readfile.c  
chmod 770 /home/guest/lab5/readfile.c



Изменяем права на файл

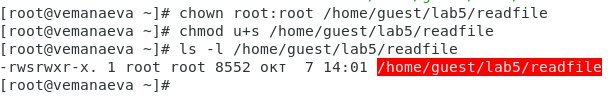
1. Проверьте, что пользователь guest не может прочитать файл readfile.c командой cat /home/guest/lab5/readfile.c (??)

Проверка

Проверка

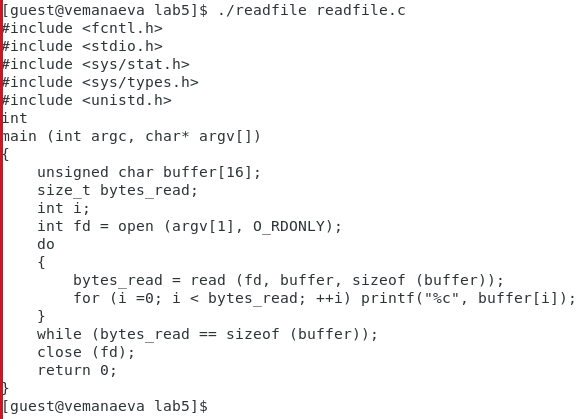
1. Смените у программы readfile владельца и установите SetUID-бит (??).

chown root:root /home/guest/lab5/readfile  
chmod u+s /home/guest/lab5/readfile



Меняем владельца

1. Проверьте, может ли программа readfile прочитать файл readfile.c (??)



Проверяем возможность чтения файла readfile.c

Да, программа может читать данный файл.

1. Проверьте, может ли программа readfile прочитать файл /etc/shadow (??)



Проверяем возможность чтения файла /etc/shadow

Да, программа может читать данный файл.

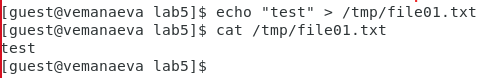
## 4.2 Исследование Sticky-бита

1. Выясним, установлен ли атрибут Sticky на директории /tmp командой ls -l / | grep tmp (??)

Sticky-бит установлен на директории /tmp

Sticky-бит установлен на директории /tmp

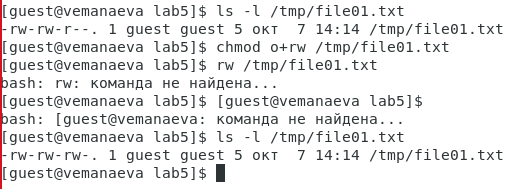
1. От имени пользователя guest создадим файл file01.txt в директории /tmpсо словом test командой echo "test" > /tmp/file01.txt (??)



Создадим файл

1. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные» командами (??):

ls -l /tmp/file01.txt  
chmod o+rw /tmp/file01.txt  
ls -l /tmp/file01.txt



Выдаём права на запись категории пользователей “все остальные”

1. От пользователя guest2 (не являющегося владельцем) попробуем прочитать файл /tmp/file01.txt командой cat /tmp/file01.txt (??)

Смотрим содержимое файла

Смотрим содержимое файла

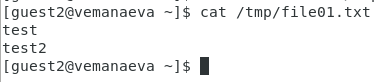
1. От пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2 командой echo "test2" >> /tmp/file01.txt (??)

Дописываем информацию в файл

Дописываем информацию в файл

Операцию выполнить удалось.

1. Проверим содержимое файла командой cat /tmp/file01.txt (??)



Смотрим содержимое файла

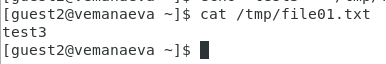
1. От пользователя guest2 попробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo "test3" > /tmp/file01.txt (??)

Перезаписываем файл

Перезаписываем файл

Операцию выполнить удалось.

1. Проверим содержимое файла командой cat /tmp/file01.txt (??)



Смотрим содержимое файла

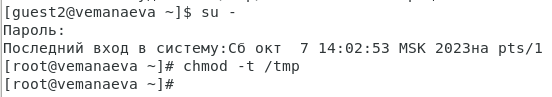
1. От пользователя guest2 попробуем удалить файл /tmp/file01.txt командой rm /tmp/fileOl.txt (??)

Пытаемся удалить файл

Пытаемся удалить файл

Файл удалить не удалось.

1. Повысим свои права до суперпользователя следующей командой su - и выполним после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp, — chmod -t /tmp (??)



Снимаем Sticky-бит с директории /tmp

1. Покинем режим суперпользователя командой exit (??)

Logout

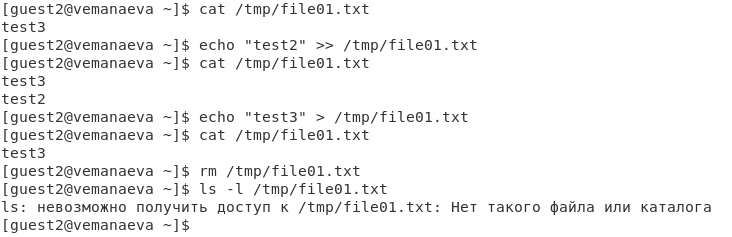
Logout

1. От пользователя guest2 проверим, что атрибута t у директории /tmp нет командой ls -l / | grep tmp (??)

Нет Sticky-бита среди атрибутов директории

Нет Sticky-бита среди атрибутов директории

1. Повторим шаги 4-9. (??)



Повторение. Получилось удалить файл, что не вышло в прошлый раз

В результате изменения удалось удалить файл от имени пользователя, не являющегося его владельцем.

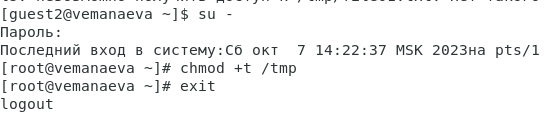
1. Удалось удалить файл от имени пользователя, не являющегося его владельцем (??).

Файл удалось удалить

Файл удалось удалить

1. Повысим свои права до суперпользователя и верните атрибут t на директорию /tmp командами (??):

su -  
chmod +t /tmp  
exit



Возвращаем атрибут директории

# 5 Выводы по проделанной работе

## 5.1 Вывод

В результате выполнения работы мы изучили механизм идентификаторов, применения SetUID- и Sticky-битов, получили практические навыки работы в консоли с дополнительными атрибутами, рассмотрели работы механизма смены идентификатора процессов пользователей, а также посмотрели влияние бита Sticky на запись и удаление файлов.

Были записаны скринкасты выполнения и защиты лабораторной работы.

Ссылки на скринкасты:

* [Выполнение, Youtube](https://youtu.be/vok8NShubfQ)
* [Выполнение, Rutube](https://rutube.ru/video/aa54f3de5171360a1705a9ccc282906d)
* [Защита презентации, Youtube](https://youtu.be/eqUCWLUuITk)
* [Защита презентации, Rutube](https://rutube.ru/video/ccf4a695cc15fe32f969f33d74a6dab3)

# 6 Список литературы

1. Колисниченко Д. Linux. От новичка к профессионалу. В подлиннике. 8-е изд. 2022. 688 с.

2. Лабораторная работа № 5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов [Электронный ресурс]. Российский Университет Дружбы Народов имени Патрису Лумумбы, 2023. URL: <https://esystem.rudn.ru/mod/resource/view.php?id=1031377>.