Отчёта по лабораторной работе №4

дисциплина: Информационная безопасность

Кашкин Иван Евгеньевич

Содержание

[Цель работы](#_Toc177202532)

[Задание](#_Toc177202533)

[Теоретическое введение](#_Toc177202534)

[Выполнение лабораторной работы](#_Toc177202535)

[Выводы](#_Toc177202536)

[Список литературы](#_Toc177202537)

# 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

## Изучение механики SetUID

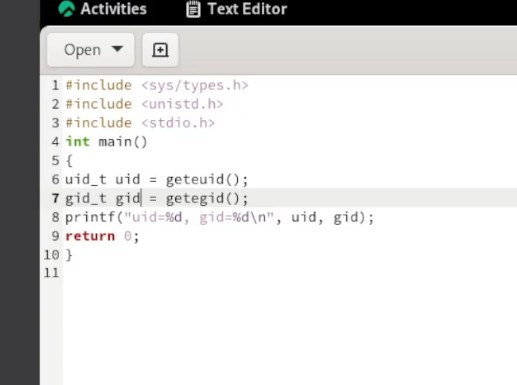
1. Вошли в систему от имени пользователя guest.
2. Написали программу simpleid.c.

#fig

1. Скомпилировали программу и убедились, что файл программы создан: gcc simpleid.c

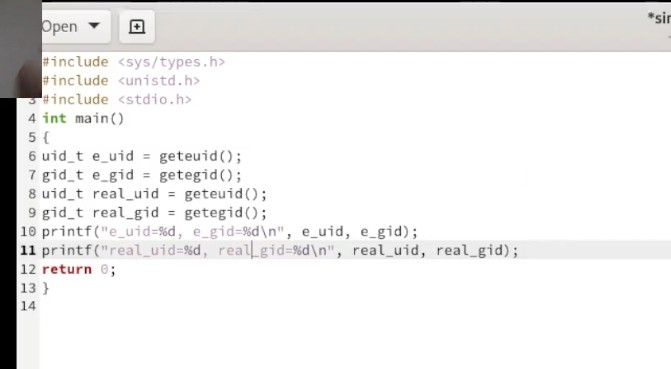
-o simpleid

1. Выполнили программу simpleid командой ./simpleid
2. Выполнили системную программу id с помощью команды id. uid и gid совпадает в обеих программах



#fig

1. Усложнили программу, добавив вывод действительных идентификаторов.



#fig

1. Скомпилировали и запустили simpleid2.c:

gcc simpleid2.c -o simpleid2

./simpleid2

1. От имени суперпользователя выполнили команды:

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

1. Использовали su для повышения прав до суперпользователя
2. Выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2:

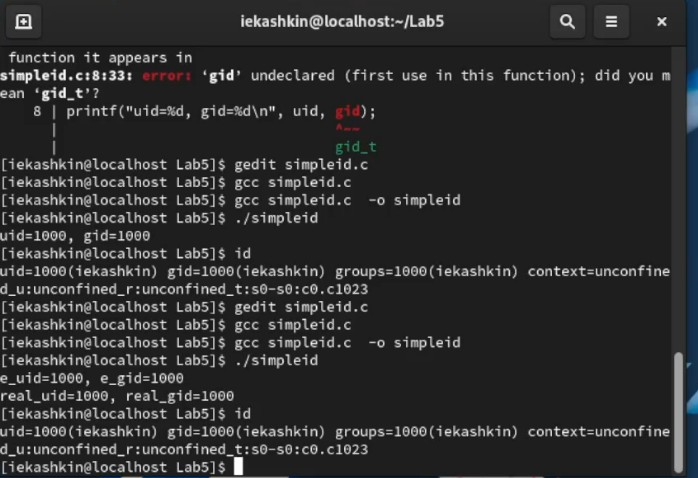
ls -l simpleid2

1. Запустили simpleid2 и id:

./simpleid2 id

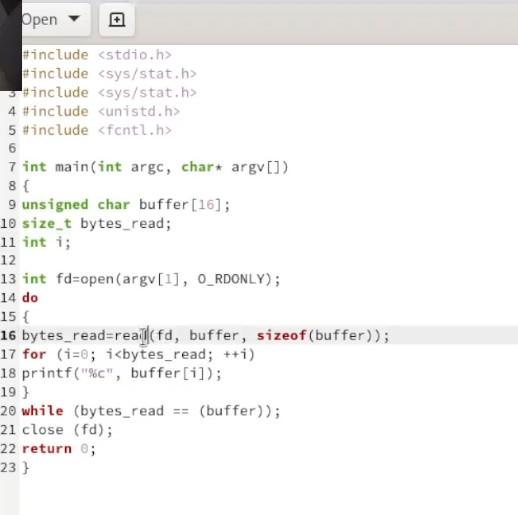
Результат выполнения программ теперь немного отличается

1. Проделали тоже самое относительно SetGID-бита.



#fig

1. Написали программу readfile.c



#fig

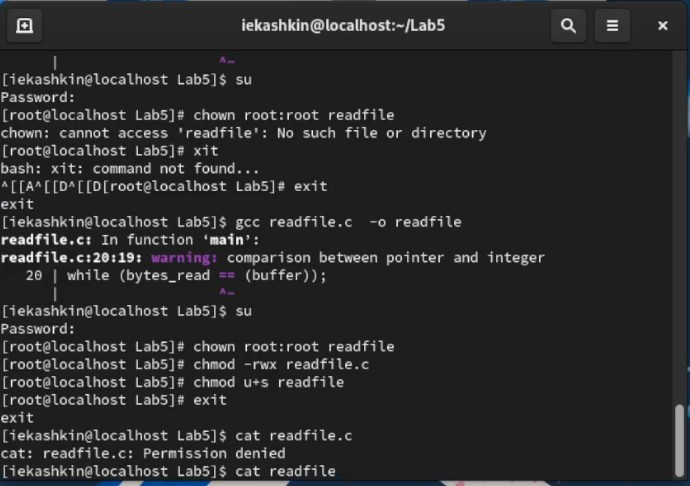
1. Откомпилировали её.

gcc readfile.c -o readfile

1. Сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог.

chown root:guest /home/guest/readfile.c chmod 700 /home/guest/readfile.c

1. Проверили, что пользователь guest не может прочитать файл readfile.c.
2. Сменили у программы readfile владельца и установили SetU’D-бит.
3. Проверили, может ли программа readfile прочитать файл readfile.c
4. Проверили, может ли программа readfile прочитать файл /etc/shadow



#fig

## Исследование Sticky-бита

1. Выяснили, установлен ли атрибут Sticky на директории /tmp:

ls -l / | grep tmp

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

echo "test" > /tmp/file01.txt

1. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные»:

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

ls -l /tmp/file01.txt

Первоначально все группы имели право на чтение, а запись могли осуществлять все, кроме «остальных пользователей».

1. От пользователя (не являющегося владельцем) попробовали прочитать файл

/file01.txt:

cat /file01.txt

1. От пользователя попробовали дозаписать в файл /file01.txt слово test3 командой:

echo "test2" >> /file01.txt

1. Проверили содержимое файла командой:

cat /file01.txt

В файле теперь записано:

Test Test2

1. От пользователя попробовали записать в файл /tmp/file01.txt слово test4, стерев при этом всю имеющуюся в файле информацию командой. Для этого воспользовалась командой echo "test3" > /tmp/file01.txt
2. Проверили содержимое файла командой

cat /tmp/file01.txt

1. От пользователя попробовали удалить файл /tmp/file01.txt командой rm /tmp/file01.txt, однако получила отказ.
2. От суперпользователя командой выполнили команду, снимающую атрибут t (Sticky- бит) с директории /tmp:

chmod -t /tmp

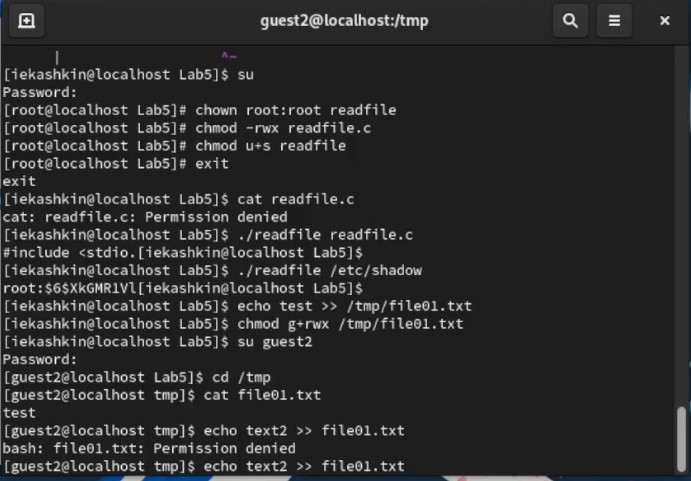
Покинули режим суперпользователя командой exit.

1. От пользователя проверили, что атрибута t у директории /tmp нет:

ls -l / | grep tmp

1. Повторили предыдущие шаги. Получилось удалить файл
2. Удалось удалить файл от имени пользователя, не являющегося его владельцем.
3. Повысили свои права до суперпользователя и вернули атрибут t на директорию /tmp

:



su

chmod +t /tmp exit

#fig

# Выводы

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Также

мы рассмотрели работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.

# Список литературы{.unnumbered}

1. [КОМАНДА CHATTR В LINUX](https://losst.ru/neizmenyaemye-fajly-v-linux)
2. [chattr](https://en.wikipedia.org/wiki/Chattr)