

# 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

## Подготовка

1. Для выполнения части заданий требуются средства разработки приложений. Проверили наличие установленного компилятора gcc командой `gcc -v`: компилятор обнаружен.
2. Чтобы система защиты SELinux не мешала выполнению заданий работы, отключили систему запретов до очередной перезагрузки системы командой `setenforce 0`:
3. Команда `getenforce` вывела Permissive:

## Изучение механики SetUID

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

Программа `simpleid`

3. Скомпилировали программу и убедились, что файл программы создан: `gcc simpleid.c -o simpleid`
4. Выполнили программу `simpleid` командой `./simpleid`
5. Выполнили системную программу `id` с помощью команды `id`. `uid` и `gid` совпадает в обеих программах

Результат работы программы `simpleid`

6. Усложнили программу, добавив вывод действительных идентификаторов.

Программа `simpleid2`

7. Скомпилировали и запустили `simpleid2.c`:

Результат работы программы `simpleid2`

```
gcc simpleid2.c -o simpleid2
./simpleid2
```

8. От имени суперпользователя выполнили команды:

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

9. Использовали `su` для повышения прав до суперпользователя

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

```
ls -l simpleid2
```

11. Запустили `simpleid2` и `id`:

```
./simpleid2
id
```

Результат работы программы simpleid2 Результат выполнения программ теперь немного отличается

- 12. Прodelали тоже самое относительно SetGID-бита.
- 13. Написали программу readfile.c(рис. @fig:007)

программа readfile

- 14. Откомпилировали её.

```
gcc readfile.c -o readfile
```

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

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

- 16. Проверили, что пользователь guest не может прочитать файл readfile.c.
- 17. Сменили у программы readfile владельца и установили SetU'D-бит.
- 18. Проверили, может ли программа readfile прочитать файл readfile.c
- 19. Проверили, может ли программа readfile прочитать файл /etc/shadow

результат программы readfile

## Исследование Sticky-бита(рис. @fig:009) (рис. @fig:010)

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

```
ls -l / | grep tmp
```

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

```
echo "test" > /tmp/file01.txt
```

- 3. 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».

- 4. От пользователя (не являющегося владельцем) попробовали прочитать файл /file01.txt:

```
cat /file01.txt
```

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

```
echo "test2" >> /file01.txt
```

- 6. Проверили содержимое файла командой:

```
cat /file01.txt
```

В файле теперь записано:

Test  
Test2

- 7. 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
- 8. Проверили содержимое файла командой

cat /tmp/file01.txt

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

chmod -t /tmp

Покинули режим суперпользователя командой exit.

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

ls -l / | grep tmp

- 12. Повторили предыдущие шаги. Получилось удалить файл
- 13. Удалось удалить файл от имени пользователя, не являющегося его владельцем.
- 14. Повысили свои права до суперпользователя и вернули атрибут t на директорию /tmp :

su  
chmod +t /tmp  
exit

результат

# 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.

- 1. [КОМАНДА CHATTR В LINUX](#)
- 2. [chattr](#)