Отчёт по лабораторной работе №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Булаев Максим Александрович НПИбд-01-19

Содержание

[1 Цель работы 1](#_Toc116142476)

[2 Выполнение лабораторной работы 1](#_Toc116142477)

[2.1 Подготовка 1](#_Toc116142478)

[2.2 Изучение механики SetUID 2](#_Toc116142479)

[2.3 Исследование Sticky-бита 6](#_Toc116142480)

[3 Выводы 8](#_Toc116142481)

[Список литературы 8](#_Toc116142482)

# 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 Выполнение лабораторной работы

## 2.1 Подготовка

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

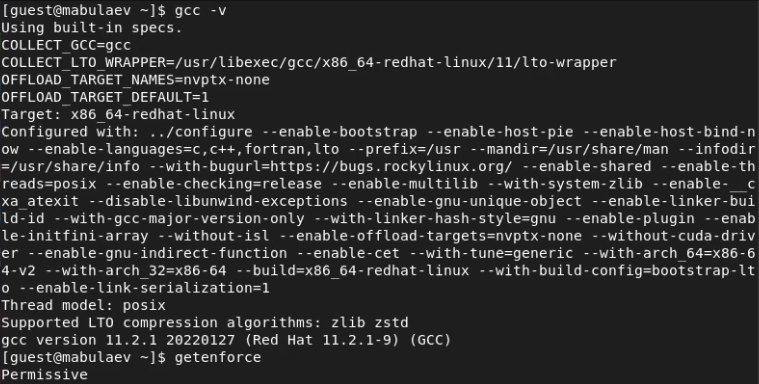


Figure 1: подготовка к работе

## 2.2 Изучение механики SetUID

1. Вошёл в систему от имени пользователя guest.
2. Написал программу simpleid.c.

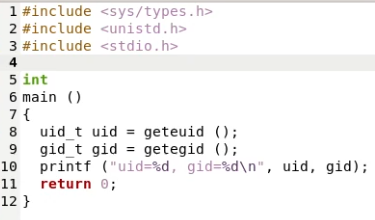


Figure 2: программа simpleid

1. Скомпилировал программу и убедился, что файл программы создан: gcc simpleid.c -o simpleid
2. Выполнил программу simpleid командой ./simpleid
3. Выполнил системную программу id с помощью команды id. uid и gid совпадает в обеих программах

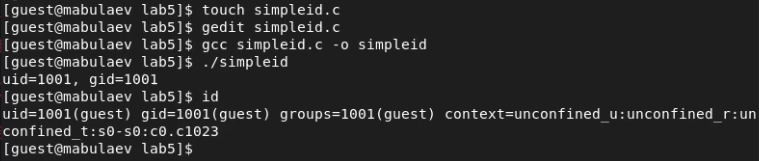


Figure 3: результат программы simpleid

1. Усложнил программу, добавив вывод действительных идентификаторов.

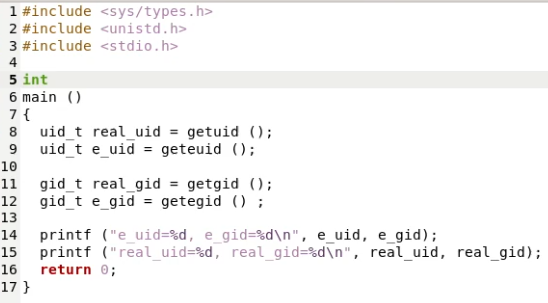


Figure 4: программа simpleid2

1. Скомпилировал и запустил simpleid2.c:

gcc simpleid2.c -o simpleid2  
./simpleid2

1. От имени суперпользователя выполнил команды:

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

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

ls -l simpleid2

1. Запустил simpleid2 и id:

./simpleid2  
id

Результат выполнения программ теперь немного отличается.

1. Проделал тоже самое относительно SetGID-бита.

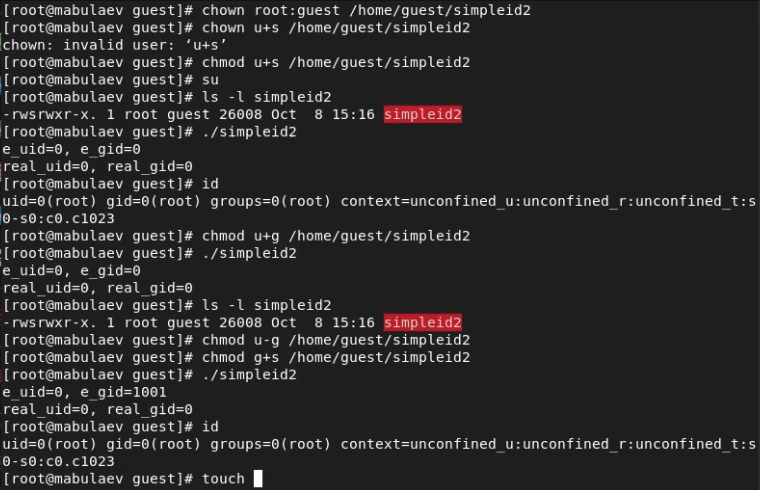


Figure 5: результат программы simpleid2

1. Написал программу readfile.c

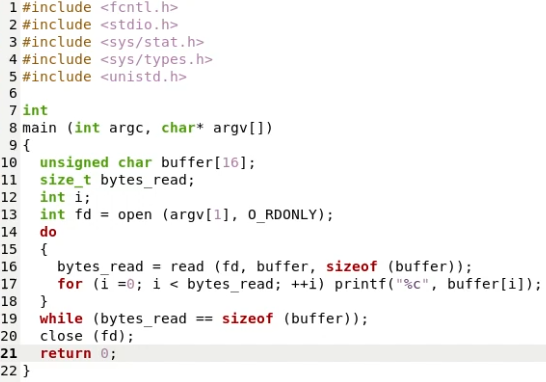


Figure 6: программа readfile

1. Откомпилировал её.

gcc readfile.c -o readfile

1. Сменил владельца у файла readfile.c и изменил 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. Проверил, что пользователь guest не может прочитать файл readfile.c.
2. Сменил у программы readfile владельца и установил SetU’D-бит.
3. Проверил, может ли программа readfile прочитать файл readfile.c
4. Проверил, может ли программа readfile прочитать файл /etc/shadow

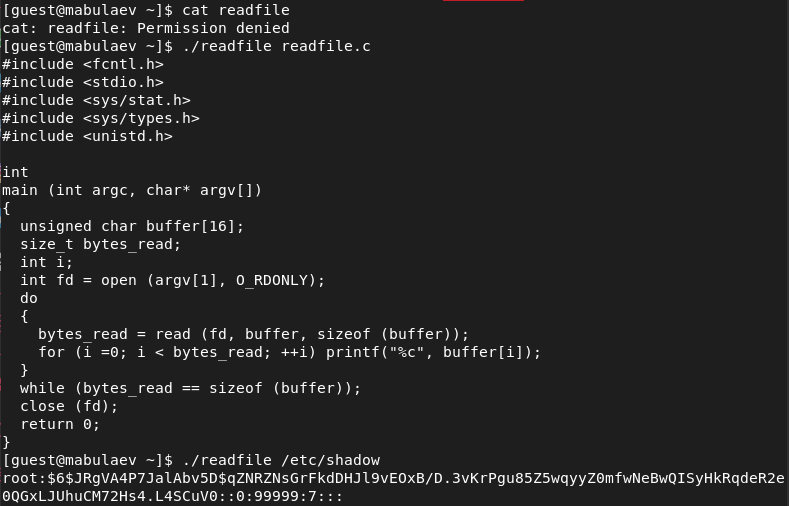


Figure 7: результат программы readfile

## 2.3 Исследование Sticky-бита

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

ls -l / | grep tmp

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

echo "test" > /tmp/file01.txt

1. Просмотрел атрибуты у только что созданного файла и разрешил 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) попробовал прочитать файл /file01.txt:

cat /file01.txt

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

echo "test2" >> /file01.txt

1. Проверил содержимое файла командой:

cat /file01.txt

В файле теперь записано:

Test  
Test2

1. От пользователя попробовал 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. Проверил содержимое файла командой

cat /tmp/file01.txt

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

chmod -t /tmp

Покинул режим суперпользователя командой exit.

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

ls -l / | grep tmp

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

su  
chmod +t /tmp  
exit

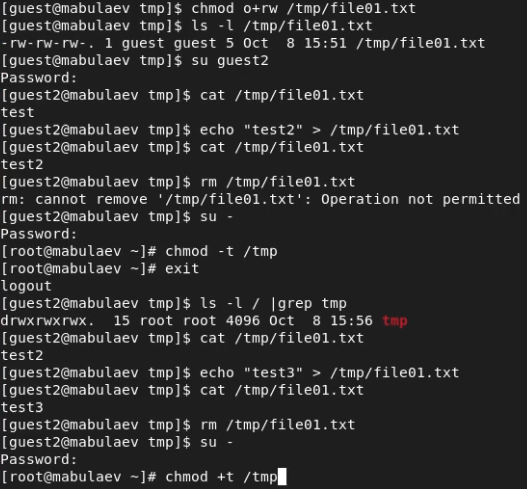


Figure 8: исследование Sticky-бита

# 3 Выводы

Таким образом, в ходе лабораторной работы я изучил механизмы изменения идентификаторов, применения SetUID- и Sticky-битов, получил практические навыки работы в консоли с дополнительными атрибутами, а также рассмотрел работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.

# Список литературы

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