Отчёт по лабораторной работе №4

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Асад Хасан НФИбд-01-18

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 Выполнение лабораторной работы

## 2.1 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:



Figure 1: подготовка к работе

## 2.2 Изучение механики SetUID

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

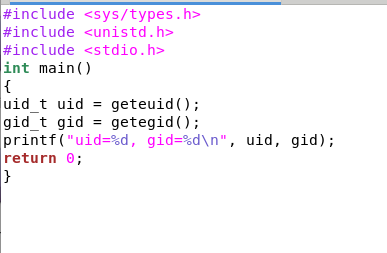


Figure 2: программа simpleid

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

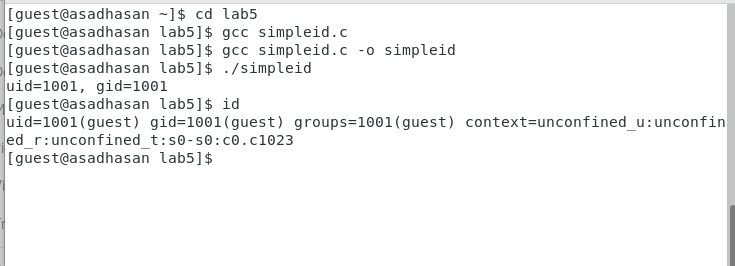


Figure 3: результат программы simpleid

1. Усложнили программу, добавив вывод действительных идентификаторов.

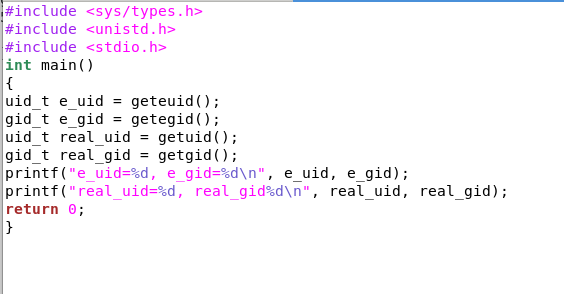


Figure 4: программа simpleid2

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.

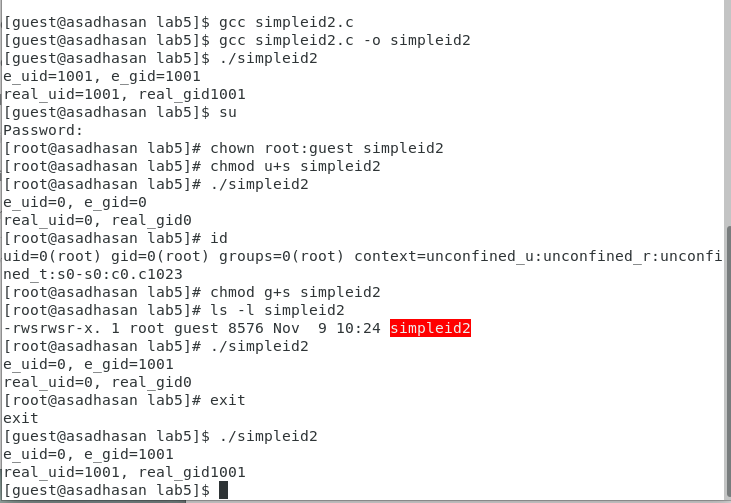


Figure 5: результат программы simpleid2

1. Написали программу readfile.c

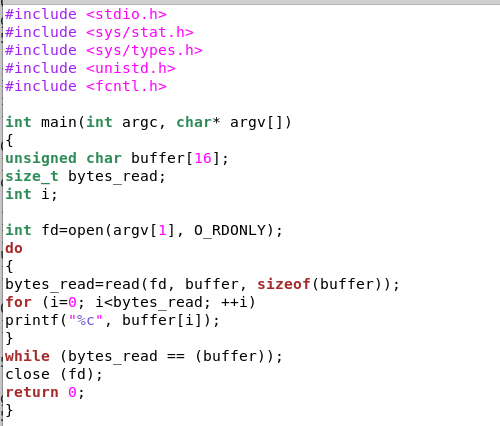


Figure 6: программа readfile

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

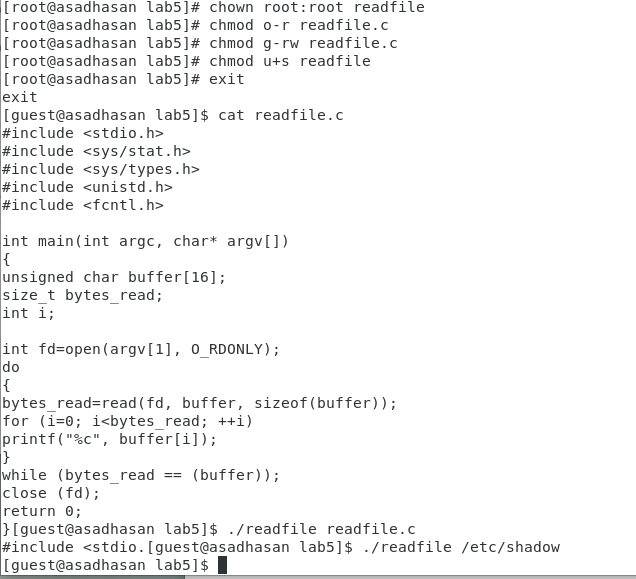


Figure 7: результат программы readfile

## 2.3 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

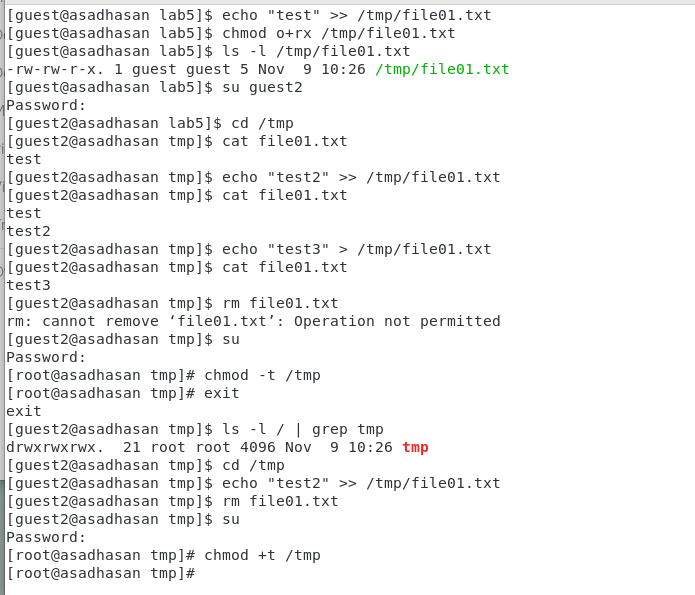


Figure 8: исследование Sticky-бита

# 3 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](https://losst.ru/neizmenyaemye-fajly-v-linux)
2. [chattr](https://en.wikipedia.org/wiki/Chattr)