Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Кувшинова К.О. группа НФИ-02-19

Содержание

# 1 Цель работы

Целью данной лабораторной работы является изучить механизмы 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 Подготовка лабораторного стенда

1. Установили компилятор gcc командой **yum install gcc**.
2. Отключили систему запретов до очередной перезагрузки системы командой **setenforce 0**. После этого команда getenforce вывела *Permissive*. (fig. 1)

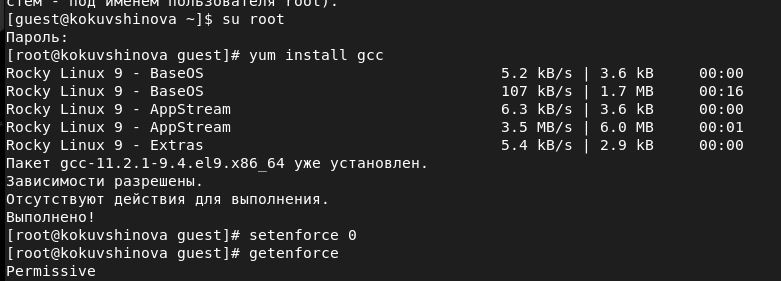


Figure 1: Установка компилятора gcc

# 3 Выполнение лабораторной работы

## 3.1 Создание программы

1. Вошли в систему от имени пользователя guest.
2. Создали программу simpleid.c. (fig. 2)



Figure 2: Программа simpleid.c

1. Скомплилировали программу и убедились, что файл программы создан с помощью команды **gcc simpleid.c -o simpleid**. Выполнили программу simpleid: **./simpleid**, а затем выполнили системную программу **id**. Обе программы выводят одинаковые значения для uid и gid. (fig. 3)

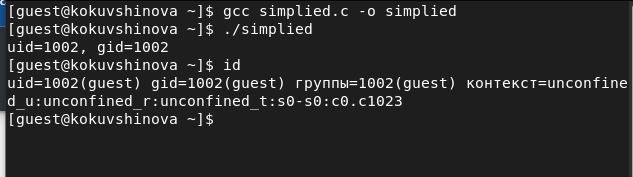


Figure 3: Выполнение программ simpleid и id

1. Усложнили программу, добавив вывод действительных идентификаторов и назвали ее simpleid2.c. (fig. 4)

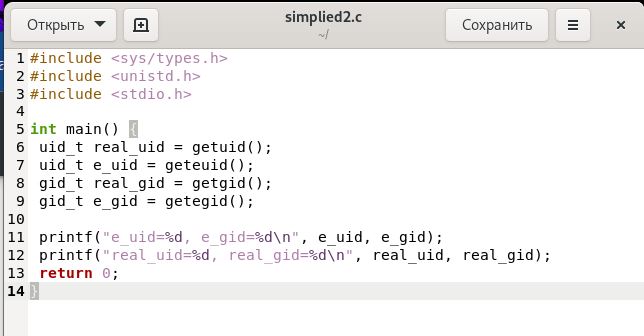


Figure 4: Программа simpleid2.c

1. Скомпилировали и запустили simpleid2.c **gcc simpleid2.c -o simpleid2** -> **./simpleid2**. (fig. 5)

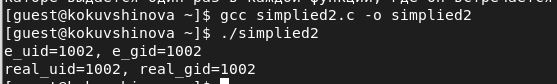


Figure 5: Выполнение программы simpleid2.c

1. От имени суперпользователя выполнили команды: **chown root:guest /home/guest/simpleid2** -> **chmod u+s /home/guest/simpleid2**. Повысили временно свои права с помощью **su root**. Выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2: **ls -l simpleid2**. Запустили simpleid2 и id с помощью команд **./simpleid2** -> **id**. Значения вывода обеих программ совпадают. (fig. 6)

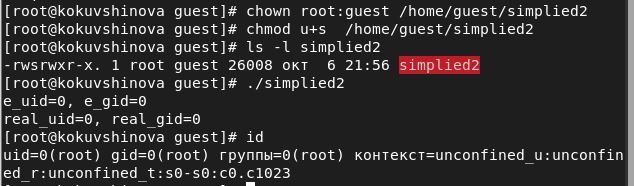


Figure 6: Выполнение программ simpleid2 и id

1. Проделали тоже самое относительно SetGID-бита.Значения вывода обеих программ совпадают, только в отличие от предыдущего пункта значение e\_gid = 1002. (fig. 7)

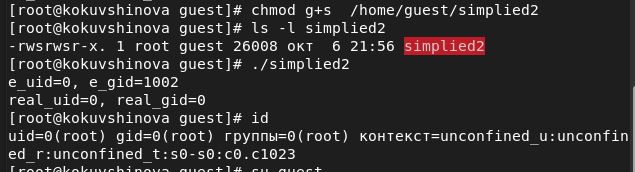


Figure 7: Выполнение программ simpleid2 и id относительно SetGID-бита

1. Создали программу readfile.c. (fig. 8)

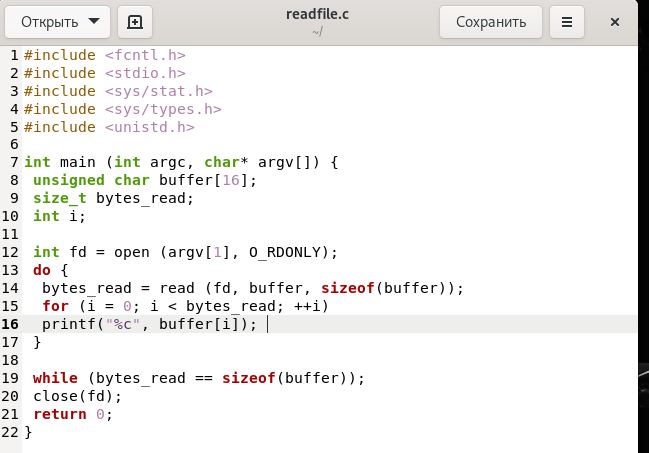


Figure 8: Программа readfile.c

1. Откомпилировали программу readfile.c: **gcc readfile.c -o readfile**. Сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог. (fig. 9)

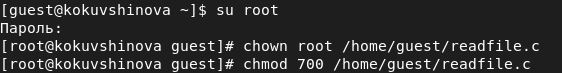


Figure 9: Смена владельца и изменение прав программы readfile.c

1. Проверили, что пользователь guest не может прочитать файл readfile.c.(fig. 10)

Figure 10: Проверка возможности чтения файла readfile.c пользователем guest

Figure 10: Проверка возможности чтения файла readfile.c пользователем guest

1. Сменили у программы readfile владельца и установили SetU’D-бит. (fig. 11)

Figure 11: Смена у программы readfile владельца и установка SetU’D-бита

Figure 11: Смена у программы readfile владельца и установка SetU’D-бита

1. Проверим, может ли программа readfile прочитать файл readfile.c. Да, может. (fig. 12)

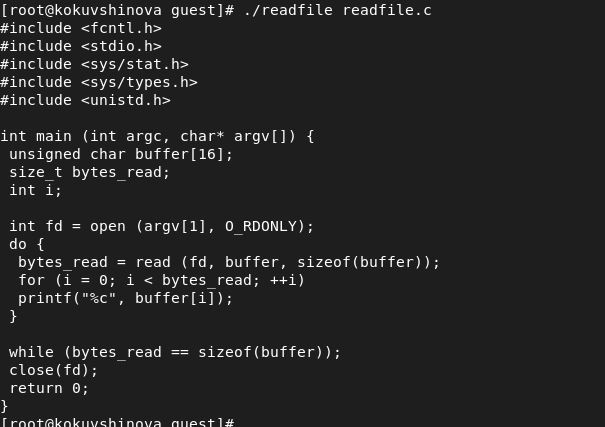


Figure 12: Чтение программой readfile файла readfile.с

1. Проверим, может ли программа readfile прочитать файл /etc/shadow. Да, может. (fig. 13)

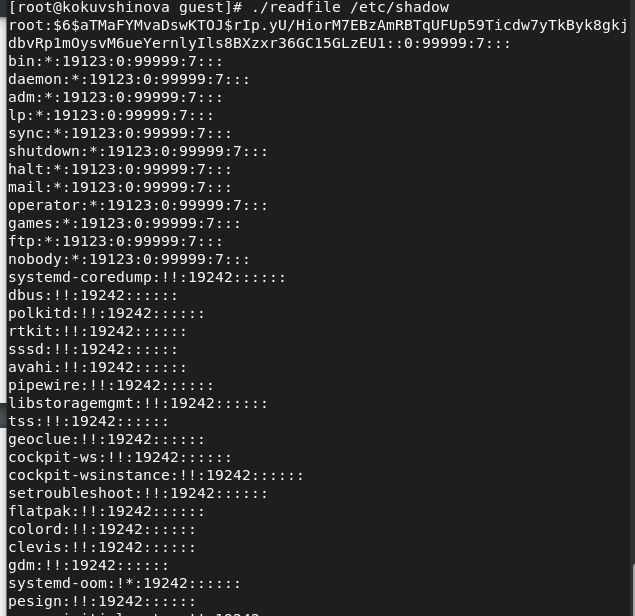


Figure 13: Чтение программой readfile файла readfile.с

## 3.2 Исследование Sticky-бита

1. Выяснили, что установлен атрибут Sticky на директории /tmp с помощью команды **ls -l / | grep tmp**. От имени пользователя guest создали файл file01.txt в директории /tmp со словом test. (fig. 14)

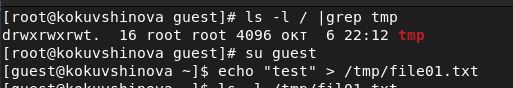


Figure 14: Выполнение команды ls -l / | grep tmp и создание файла file01.txt

1. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные». (fig. 15)

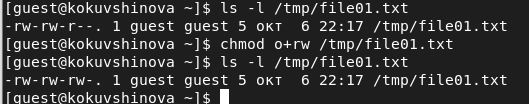


Figure 15: Арибуты файла file01.txt

1. От пользователя guest2 (не являющегося владельцем) попробовали прочитать, дозаписать, записать, удалить файл /tmp/file01.txt. Выполнено все, кроме удаления файла. (fig. 16)

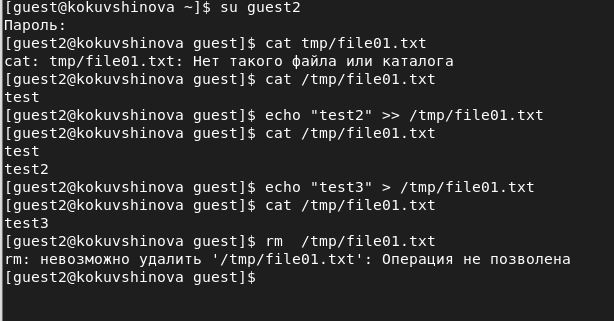


Figure 16: Чтение, дозапись, запись, удаление файл /tmp/file01.txt

1. Повысили свои права до суперпользователя командой **su -** и выполнили после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: **chmod -t /tmp**. Покинули режим суперпользователя командой **exit**.(fig. 17)

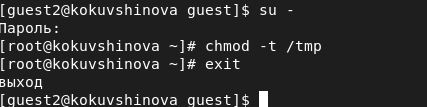


Figure 17: Сняли Sticky-бит с директории /tmp

1. От пользователя guest2 проверили, что атрибута t у директории /tmp нет. (fig. 18)

Figure 18: Отсутсвие атрибута t у директории /tmp

Figure 18: Отсутсвие атрибута t у директории /tmp

1. Повторили предыдущие шаги. В данном случае получилось выполнить удаление файла. (fig. 19)

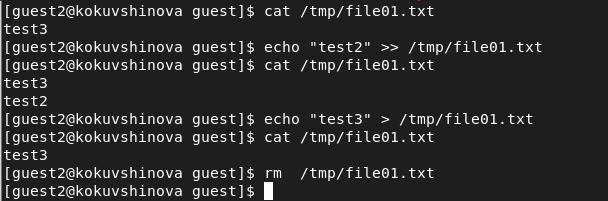


Figure 19: Чтение, дозапись, запись, удаление файл /tmp/file01.txt без атрибута t

1. Повысили свои права до суперпользователя и вернули атрибут t на директорию /tmp. (fig. 20)

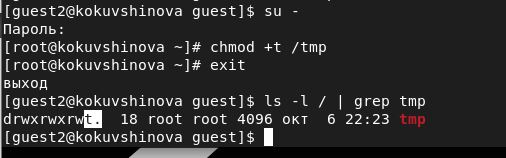


Figure 20: Чтение, дозапись, запись, удаление файл /tmp/file01.txt без атрибута t

# 4 Вывод

В ходе выполнения лабораторной работы мы приобрели изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Рассмотрели работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

# 5 Библиография

1. Кулябов Д. С., Королькова А. В., Геворкян М. Н. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов [Текст] / Кулябов Д. С., Королькова А. В., Геворкян М. Н. - Москва: - 7 с. [^1]: Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов.
2. Справочник 70 основных команд Linux: полное описание с примерами (https://eternalhost.net/blog/sozdanie-saytov/osnovnye-komandy-linux)