Лабораторная работа №5

Лукьянова Ирина Владимировна, НФИбд-02-19

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.[[1]](#footnote-20)

# 2 **Выполнение лабораторной работы**

От имени суперпользователя устанавливаем gcc командой yum install gcc(рис. 1)

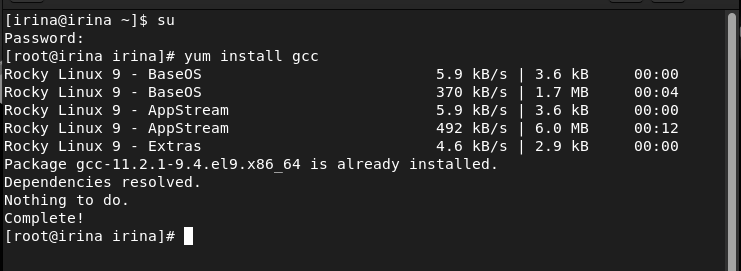


Figure 1: Установка gcc

Отключаем систему запретов до очередной перезагрузки системы командой setenforce 0. После этого команда getenforce выводит Permissive.(рис. 2)

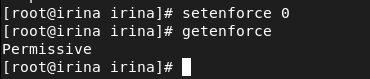


Figure 2: Отключение системы запретов

Компилятор языка С называется gcc. Компилятор языка С++ называется g++ и запускается с параметрами почти так же, как gcc. Проверяем это следующими командами:(рис. 3)

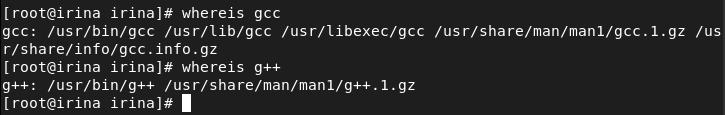


Figure 3: Компиляторы С и С++

Входим в систему от имени пользователя guest и создаем программу simpleid.c:(рис. 4), (рис. 5)

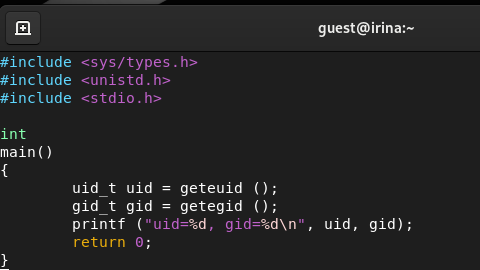


Figure 4: Программа simpleid.c

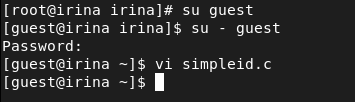


Figure 5: Вход и создание программы

Компилируем и выполняем программу simpleid. После выполняем программу id.(рис. 6).

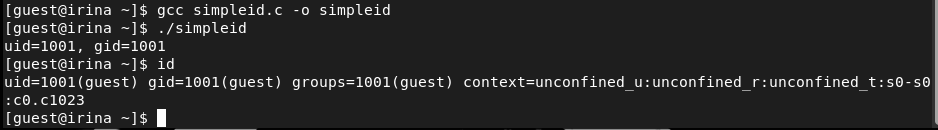


Figure 6: Выполнение программ

Усложняем программу simpleid2.c, добавив вывод действительных идентификаторов.(рис. 7).

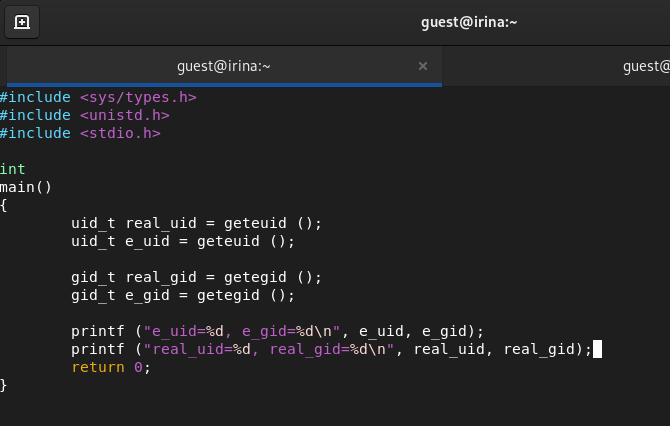


Figure 7: Программа simpleid2.c

Компилируем и запускаем simpleid2.c. (рис. 8)

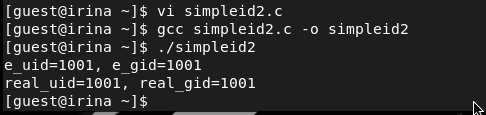


Figure 8: Компилируем и запускаем simpleid2.c

От имени суперпользователя выполняем команды:(рис. 9)

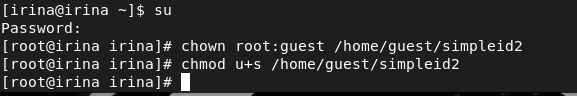


Figure 9: Выполнение команд

Используйте sudo или повысьте временно свои права с помощью su. Поясните, что делают эти команды. Команда sudo позволяет пользователям выполнять указанные программы с административными привилегиями без ввода пароля суперпользователя root.

Выполняем проверку правильности установки новых атрибутов и смены владельца файла simpleid2. После запускаем simpleid2 и id(рис. 10)

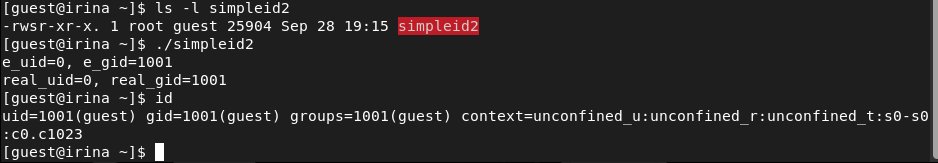


Figure 10: Выполняем проверку

Проделываем тоже самое относительно SetGID-бита.(рис. 11)

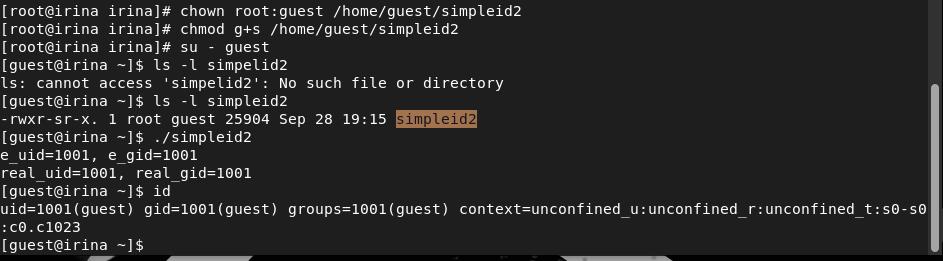


Figure 11: Повтор операций для SetGID-бита

Создаем программу readfile.c(рис. 12)

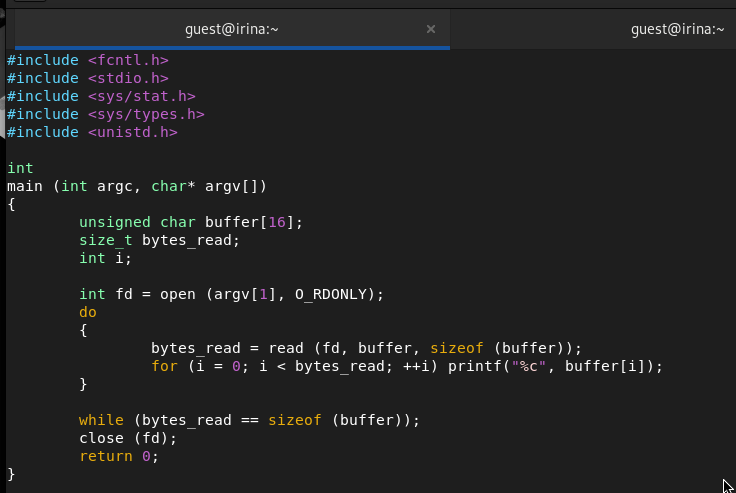
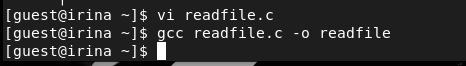


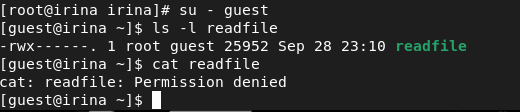
Figure 12: Создаем программу readfile.c

Компилируем эту программу(рис. **¿fig:013?**) 

Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь мог прочитать его, a guest не мог.(рис. 13)

Figure 13: Меняем владельца и права файла

Figure 13: Меняем владельца и права файла

Проверяем, что пользователь guest не может прочитать файл readfile.c.(рис. **¿fig:015?**) 

Сменим у программы readfile владельца и установим SetU’D-бит.(рис. 14)

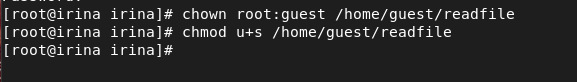


Figure 14: Меняем владельца

Проверяем, может ли программа readfile прочитать файл readfile.c (рис. 15)

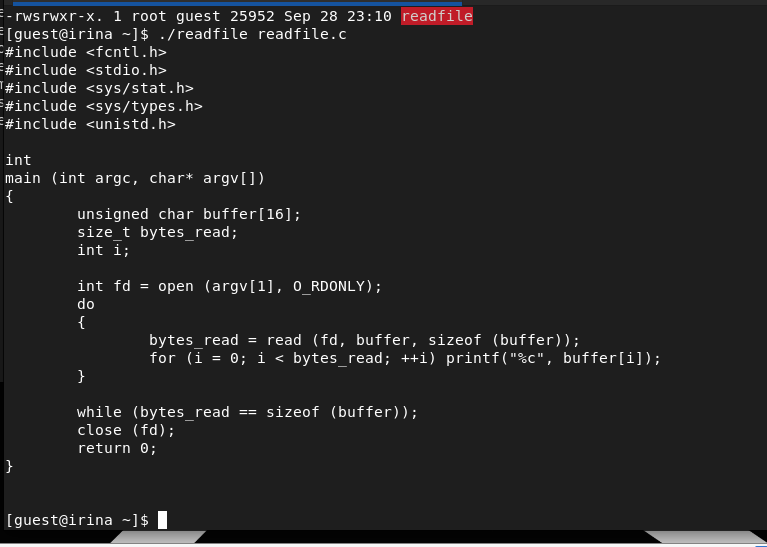


Figure 15: Проверка

Проверяем, может ли программа readfile прочитать файл /etc/shadow (рис. 16)

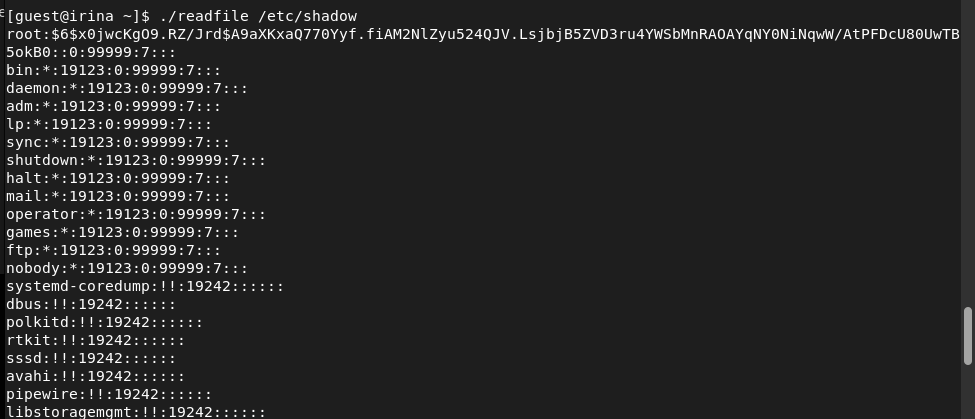


Figure 16: Проверка

## 2.1 **Исследование Sticky-бита**

Выясняем, установлен ли атрибут Sticky на директории /tmp. После от имени пользователя guest создаем файл file01.txt в директории /tmp со словом test.

Смотрим атрибуты у созданного файла и разрешаем чтение и запись для категории пользователей «все остальные». (рис. 17), (рис. 18)

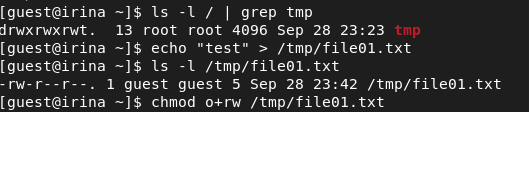


Figure 17: Работа с атрибутами

Figure 18: Проверка

Figure 18: Проверка

От пользователя guest2 (не являющегося владельцем) пробуем прочитать файл /tmp/file01.txt Далее от пользователя guest2 пробуем дозаписать в файл /tmp/file01.txt слово test2 и проверяем содержимое файла(рис. 19)

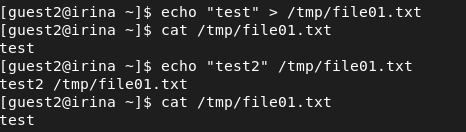


Figure 19: Пробуем прочитать файл

От пользователя guest2 пробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию и проверяем содержимое файла.(рис. 20)



Figure 20: Работа с файлом

От пользователя guest2 пробуем удалить файл /tmp/file01.txt(рис. 21)

Figure 21: Попытка удалить файл

Figure 21: Попытка удалить файл

Повышаем свои права до суперпользователя и выполняем после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp:(рис. 22)

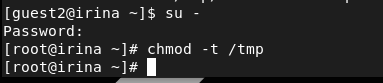


Figure 22: Работа с атрибутами

Покидаем режим суперпользователя командой exit и от пользователя guest2 проверяем, что атрибута t у директории /tmp нет:(рис. 23)

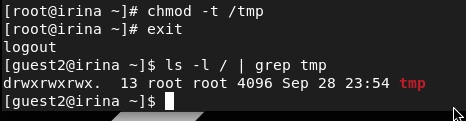


Figure 23: Проверка

Повторяем предыдущие шаги.(рис. 24)

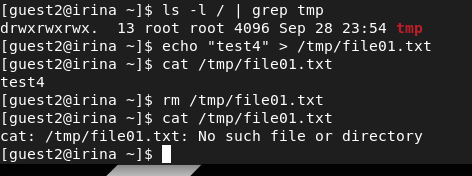


Figure 24: Повтор команд

Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Да, удалось. (рис. 25)

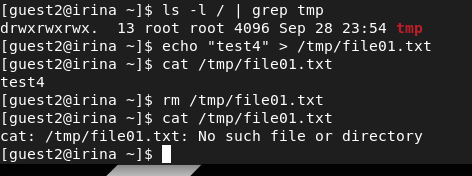


Figure 25: Попытка удалить файл №2

Повышаем свои права до суперпользователя и возвращаем атрибут t на директорию /tmp: (рис. 26)

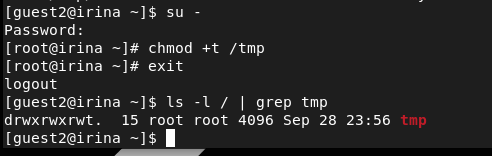


Figure 26: Работа с атрибутами

# 3 Выводы

В ходе выполнения данной лабораторной работы я изучила механизмы изменения идентификаторов и применения SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

# 4 Список литературы

1. [Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов. / Кулябов Д. С., Королькова А. В., Геворкян М. Н. - Москва: - 7 с.](https://esystem.rudn.ru/pluginfile.php/1651889/mod_resource/content/2/005-lab_discret_sticky.pdf)
2. [Руководство по оформлению Markdown.](https://gist.github.com/Jekins/2bf2d0638163f1294637)

1. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов. [↑](#footnote-ref-20)