Отчёт лабораторной работы 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Турсунов Баходурхон Азимджонович

Содержание

# 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.

## 2.1 Подготовка лабораторного стенда

1. С помощью команды gcc -v проверил, установлен ли компилятор gcc в моей системе, как оказалось не установлен

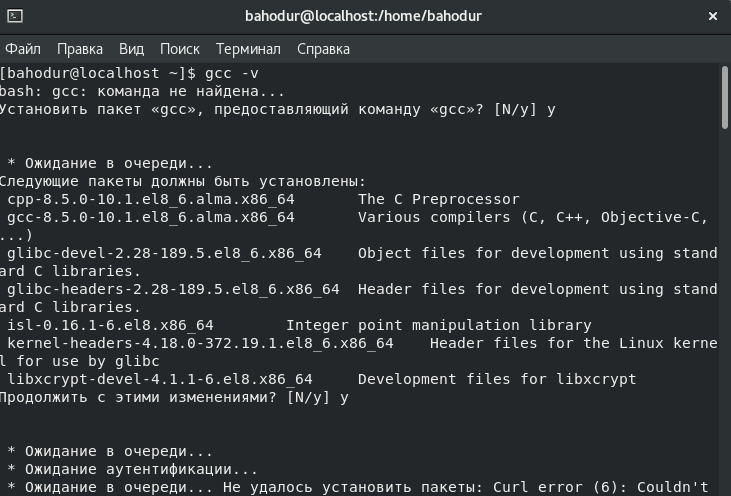


Figure 1: gcc

1. После установления компилятора gcc с помощью команды gcc -v узнаем подробности и версию нашего компилятора, затем отключил систему запретов до очередной перезагрузки системы командой setenforce 0, после этого команда getenforce вывела Permissive.

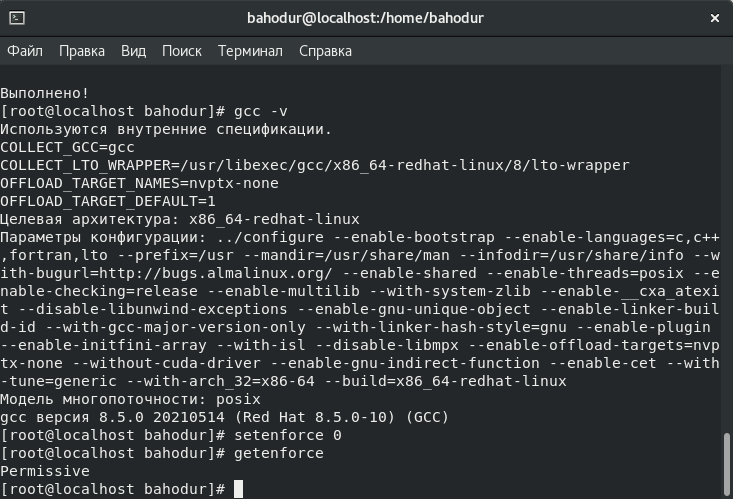


Figure 2: gcc -v

## 2.2 Изучение механики SetUID

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



Figure 3: simpleid.c

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

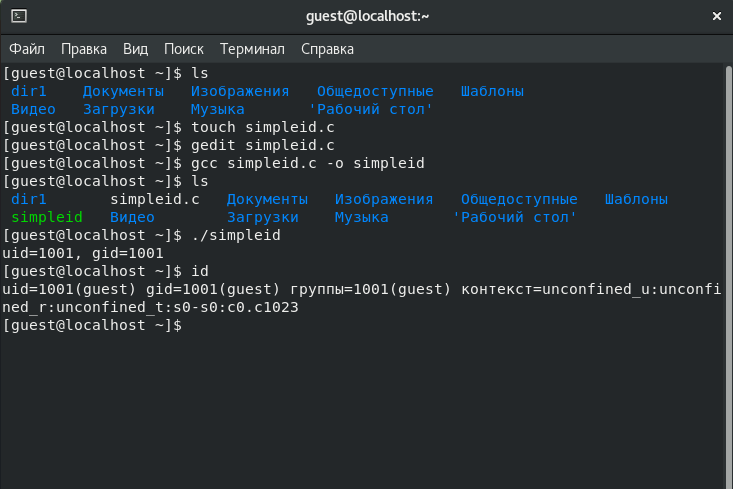


Figure 4: результат программы simpleid

1. Усложнил программу, добавив вывод действительных идентификаторов

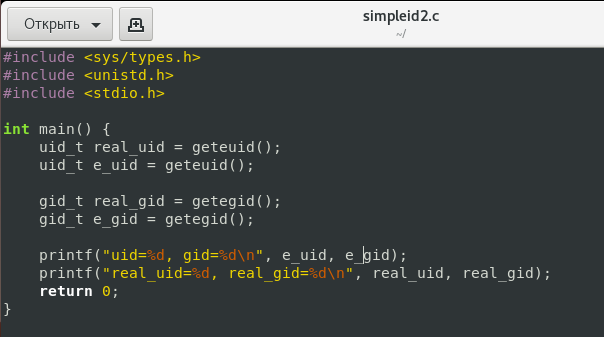


Figure 5: программа simpleid2

1. Далее скомпилировал и запустил simpleid2.c и от имени суперпользователя выполнил команды chown и chmod u+s и выполнил проверку правильности установки новых атрибутов и имени владельца файла

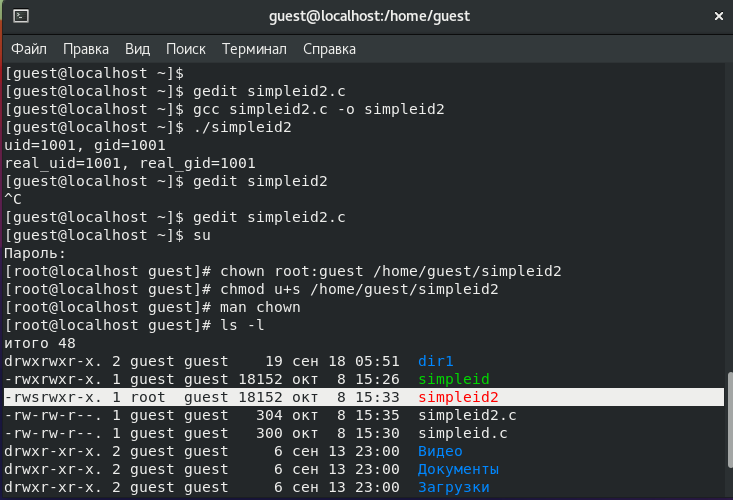


Figure 6: результат программы simpleid2

1. Написал программу readfile.c

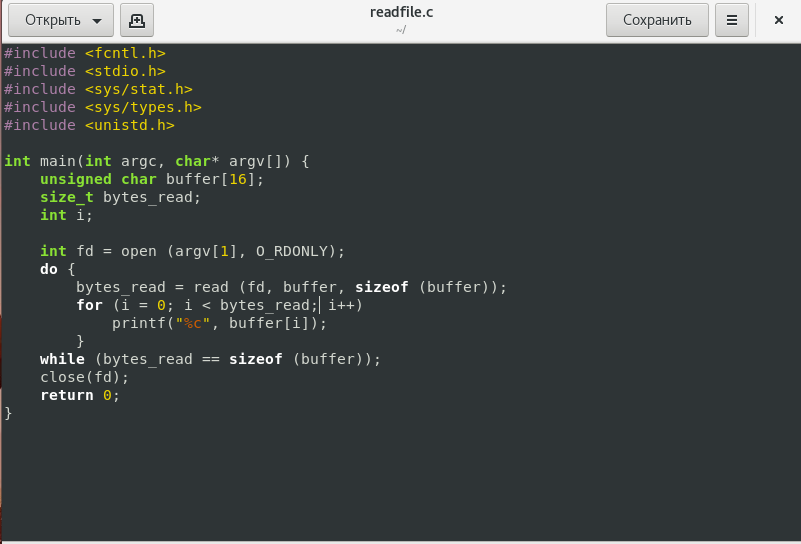


Figure 7: программа readfile.c

1. Скомпилировал файл



Figure 8: результат программы readfile.c

1. Сменил владельца у файла и изменил права так, чтобы только суперпользователь мог прочитать его, а guest не мог. И проверил что пользователю отказано в доступе и он не может прочитать файл.

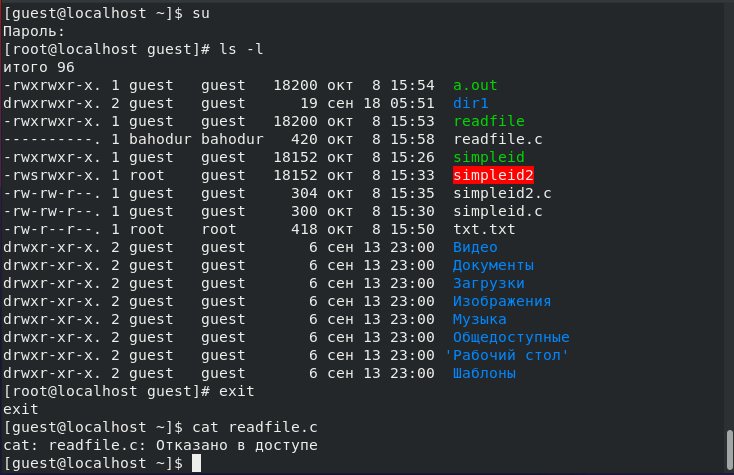


Figure 9: результат программы readfile.c

## 2.3 Исследование Sticky-бита

1. Выяснил, установлен ли атрибут Sticky на директории /tmp.
2. От имени пользователя guest создал файл file01.txt в директории /tmp со словом test.
3. Просмотрел атрибут у только что созданного файла и разрешил чтение и запись для категории пользователей “все остальные”

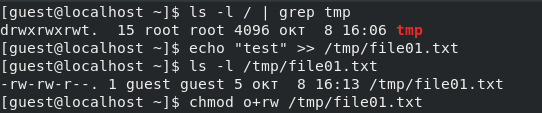


Figure 10: исследование Sticky-бита

* Первоначально все группы имели право на чтение, а запись могли осуществлять все, кроме “остальных пользователей”

Figure 11: исследование Sticky-бита

Figure 11: исследование Sticky-бита

1. От пользователя (не являющегося владельцем) попробовал прочитать файл /file01.txt
2. От пользователя попробовал записал в файл test и test2 и проверил командой cat /tmp/file01.txt

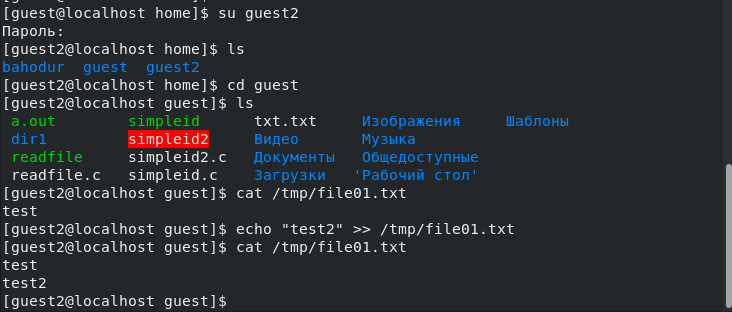


Figure 12: чтение и запись на файл

* И убеждаемся, что в файле действительно есть наши записанные слова.

1. От пользователя попробовал записать в файл слово test3, стерев при этом всю имеющуюся в файле информацию. И командой echo “test3” >> /tmp/file01.txt записал новую информацию в файл
2. Также проверил информацию, которая лежит в файле командой cat /tmp/file01.txt
3. От пользователя попробовал удалить файл, и получил отказ
4. От имени суперпользователя выполнил команду, снимающую атрибут t с директории /tmp и покинул режим суперпользователя
5. Повторил предыдущие шаги и мне удалось удалить файл

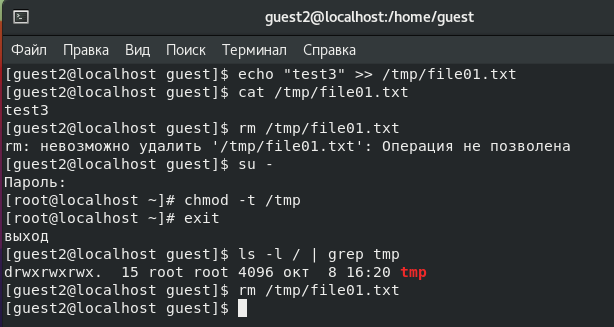


Figure 13: чтение и запись на файл

# Вывод

* Изучил механизмы изменение идентификаторов, применения SetUID- и Sticky-битов. Получил практические навыки работы в консоли с дополнительными атрибутами. Также рассмотрел работу механизма смены идентификатора процессов пользователей и влияние бита на запись и удаление файлов.