Лабораторная работа №5

Информационная безопасность

Маметкадыров Ынтымак | НПМбд-02-20

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.

# Теоретическое введение

SetUID, SetGID и Sticku — это специальные типы разрешений, которые позволяют задавать расширенные права доступа на файлы и каталоги.

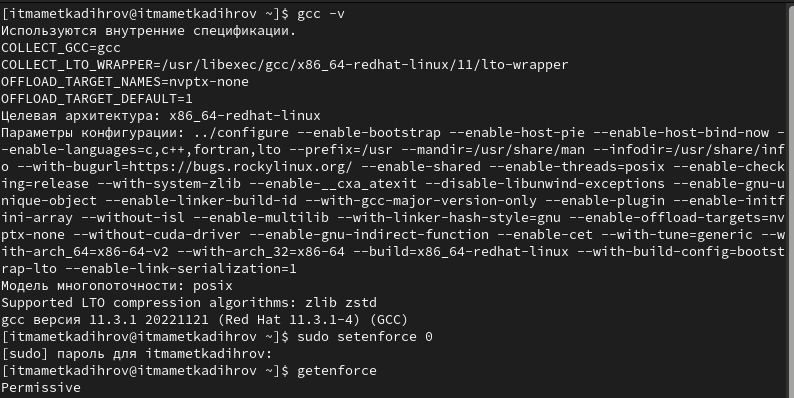
* SetUID — это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволят поднять привилегии пользователя в случае, если это необходимо. Наличие SetUID бита выражается в том, что на месте классического бита x выставлен специальный бит s: -rwsr-xr-x
* SetGID — очень похож на SetUID с отличием, что файл будет запускаться от имени группы, который владеет файлом: -rwxr-sr-x
* Sticky — в случае, если этот бит установлен для папки, то файлы в этой папке могут быть удалены только их владельцем. Наличие этого бита показывается через букву t в конце всех прав: drwxrwxrwxt

Более подробно см. в [@prava:bash].

# Выполнение лабораторной работы

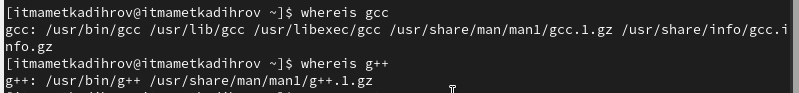
## Создание программы

Для начала убедились, что компилятор gcc установлен, исолпьзуя команду “gcc -v”. Затем отключили систему запретов до очередной перезагрузки системы командой “sudo setenforce 0”, после чего команда “getenforce” вывели “Permissive” (рис. [-@fig:001])



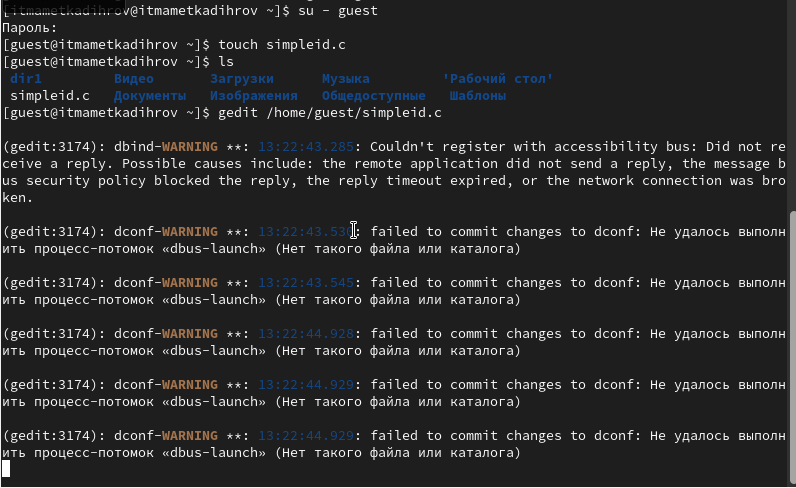
Предварительная подготовка

Проверили успешное выполнение команд. (рис. [-@fig:002])



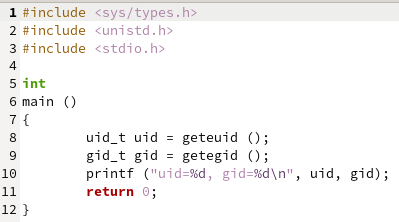
Команда “whereis”

Вошли в систему от имени пользователя guest командой “su - guest”. Создали программу simpleid.c командой “touch simpleid c” и открыли её в редакторе командой “gedit /home/guest/simpleid.c” (рис. [-@fig:003])



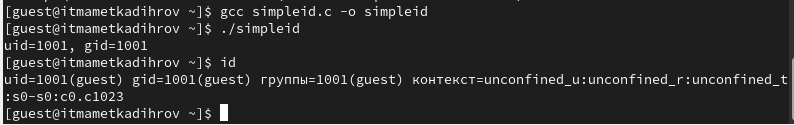
Вход в систему и создание программы

Код программы выглядит следующим образом (рис. [-@fig:004]).



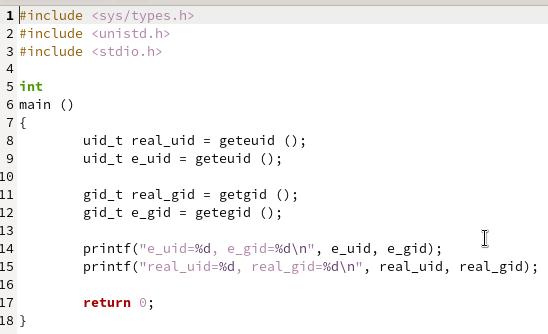
Код программы simpleid.c

Скомпилировали программу и убедились, что файл программы был создан командой “gcc simpleid.c -o simpleid”. Выполнили программу simpleid командой “./simpleid”, а затем выполнили системную программу id командой “id”. Результаты, полученные в результате выполнения обеих команд, совпадают (uid=1001 и gid=1001) (рис. [-@fig:005]).



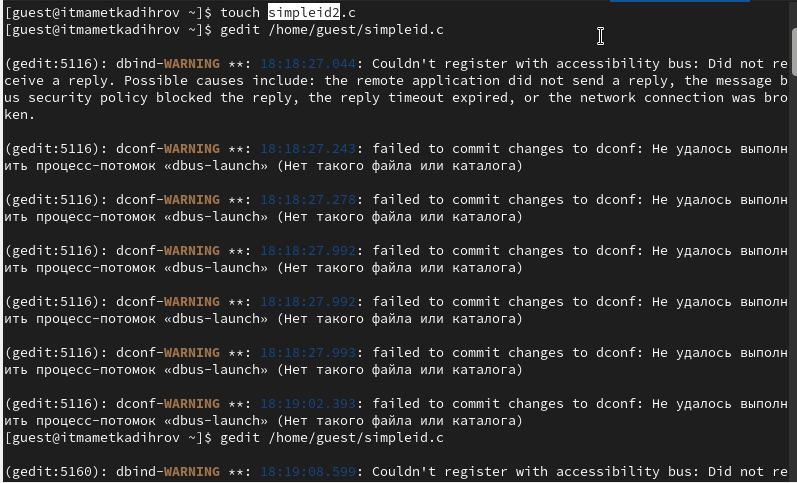
Компиляция и выполнение программы simpleid

Усложнили программу, добавив вывод действительных идентификаторов (рис. [-@fig:006]).



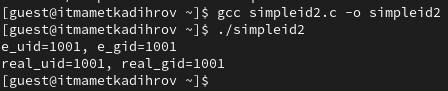
Усложнение программы

Получившуюся программу назвали simpleid2.c (рис. [-@fig:007]).



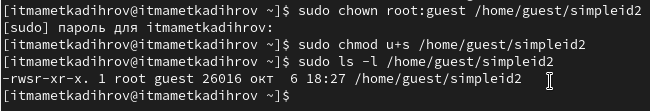
Создание файла simpleid2.c

Скомпилировали и запустили simpleid2.c командами “gcc simpleid2.c -o sipleid2” и “./simpleid2” (рис. [-@fig:008]).



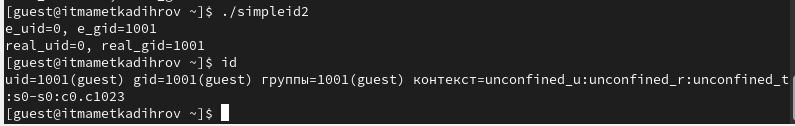
Компиляция и выполнение программы simpleid2

От имени суперпользователя выполнили команды “sudo chown root:guest /home/guest/simpleid2” и “sudo chmod u+s /home/guest/simpleid2”, затем выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой “sudo ls -l /home/guest/simpleid2” (рис. [-@fig:009]). Этими командами была произведена смена пользователя файла на root и установлен SetUID-бит.



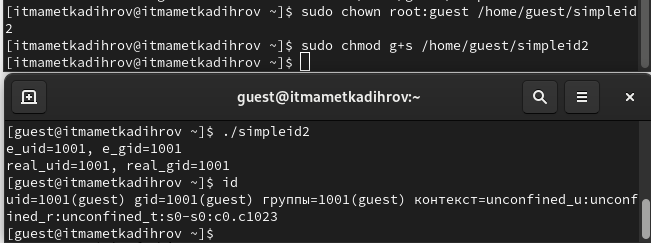
Установка новых атрибутов (SetUID) и смена владельца файл

Запустили программы simpleid2 и id. Теперь появились различия в uid (рис. [-@fig:010])



Запуск simpleid2 после установки SetUID

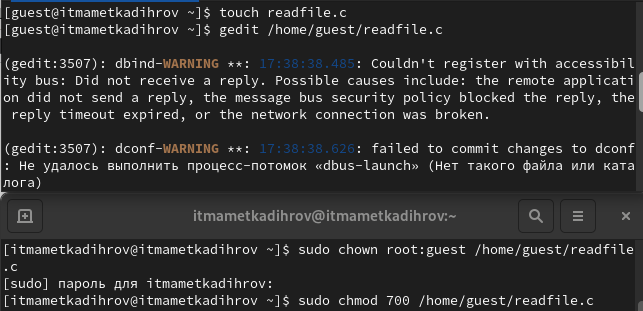
Проделали тоже самое относительно SetGID-бита. Также можем заметить различия с предыдущим пунктом (рис. [-@fig:011]).



Запуск simpleid2 после установки SetGID

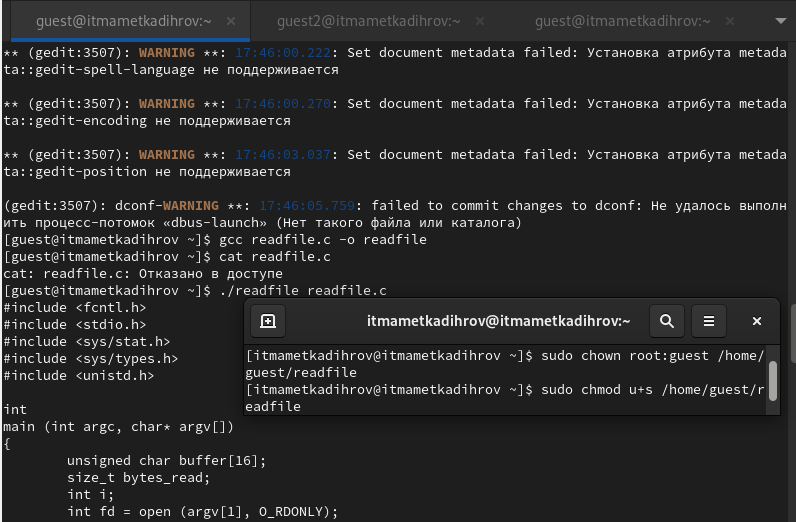
Создаем программу readfile.c.

Скомпилировали созданную программу командой “gcc readfile.c -o readfile”. Сменили владельца у файла readfile.c командой “sudo chown root:guest /home/guest/readfile.c” и поменяли права так, чтобы только суперпользователь мог прочитать его, а guest не мог, с помощью команды “sudo chmod 700 /home/guest/readfile.c”. Теперь убедились, что пользователь guest не может прочитать файл readfile.c командой “cat readfile.c”, получив отказ в доступе (рис. [-@fig:012]).



Смена владельца и прав доступа у файла readfile.c

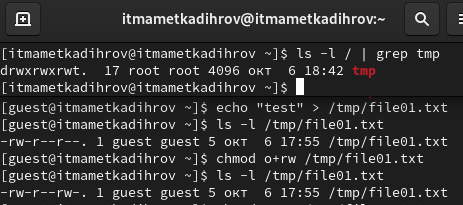
Поменяла владельца у программы readfile и устанавила SetUID. Проверила, может ли программа readfile прочитать файл readfile.c командой “./readfile readfile.c”. Прочитать удалось.Аналогично проверила, можно ли прочитать файл /etc/shadow. Прочитать удалось (рис. [-@fig:013]).



Запуск программы readfile

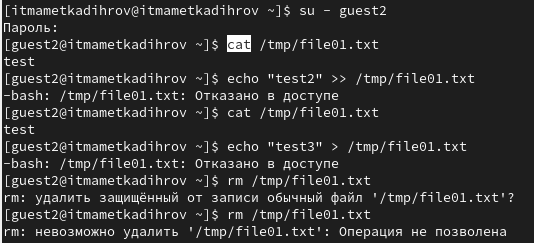
## 3.2 Исследование Sticky-бита

Командой “ls -l / | grep tmp” убеждилась, что атрибут Sticky на директории /tmp установлен. От имени пользователя guest создала файл file01.txt в директории /tmp со словом test командой “echo”test” > /tmp/file01.txt”. Просматрела атрибуты у только что созданного файла и разрешаем чтение и запись для категории пользователей “все остальные” командами “ls -l /tmp/file01.txt” и “chmod o+rw /tmp/file01.txt” (рис. [-@fig:014]).



Создание файла file01.txt

От имени пользователя guest2 попробовали прочитать файл командой “cat /tmp/file01.txt” - это удалось. Далее попытались дозаписать в файл слово test2, проверить содержимое файла и записать в файл слово test3, стерев при этом всю имеющуюся в файле информацию - эти операции удалось выполнить только в случае, если еще дополнительно разрешить чтение и запись для группы пользователей командой “chmod g+rw /tmp/file01.txt”. От имени пользователя guest2 попробовали удалить файл - это не удается ни в каком из случаев, возникает ошибка (рис. [-@fig:015]).



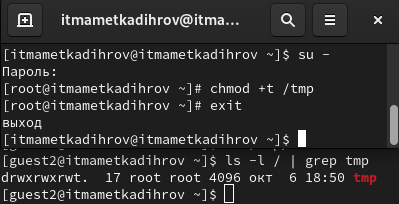
Попытка выполнить действия над файлом file01.txt от имени пользователя guest2

Повысили права до суперпользователя командой “su -” и выполнили команду, снимающую атрибут t с директории /tmp “chmod -t /tmp”. После чего покинули режим суперпользователя командой “exit”. Повторили предыдущие шаги. Теперь мне удалось удалить файл file01.txt от имени пользователя, не являющегося его владельцем (рис. [-@fig:016]).



Удаление атрибута t (Sticky-бита) и повторение действий

Повысили свои права до суперпользователя и вернули атрибут t на директорию /tmp (рис. [-@fig:017]).



Возвращение атрибута t (Sticky-бита)

# Выводы

В ходе выполнения данной лабораторной работы мы изучили механизмы изменения идентификаторов, применение 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.

# Список литературы