Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Artem Smirnov"

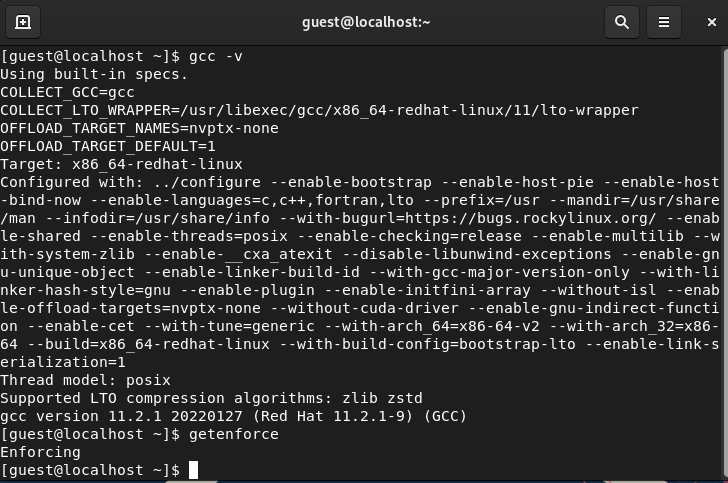
Содержание

# 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.

# Выполнение лабораторной работы

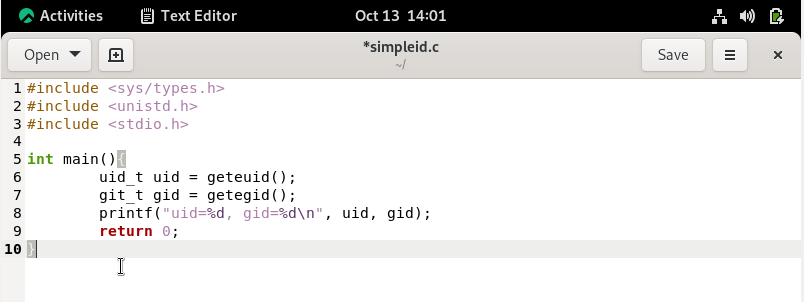
Проверю, установлен ли у меня компилятор gcc командой gcc –v.



##### Создание программы

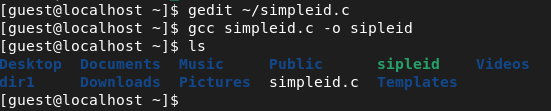
Войду в систему от имени пользователя guest.

Создам программу simpleid.c.



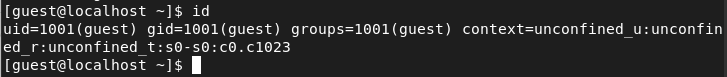
Скомпилирую программу командой gcc simpleid.c -o simpleid и удостоверюсь, что  
файл программы создан

Выполню программу simpleid командой ./simpleid

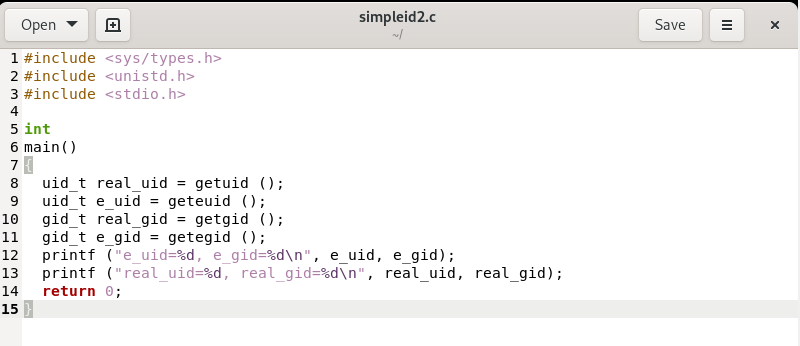




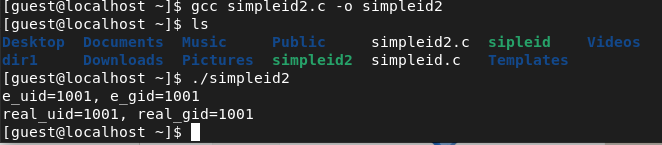
Выполню системную программу id командой id. Результат совпадает.



Усложню программу, добавив вывод действительных идентификаторов. Создам  
новый файл simpleid2.c



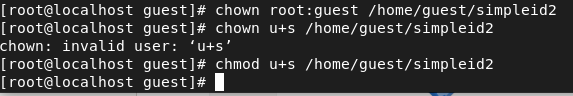
Скомпилирую и запущу simpleid2.c



##### Работа с е SetUID-битом

От имени суперпользователя выполню команды:  
chown root:guest /home/guest/simpleid2  
chmod u+s /home/guest/simpleid2



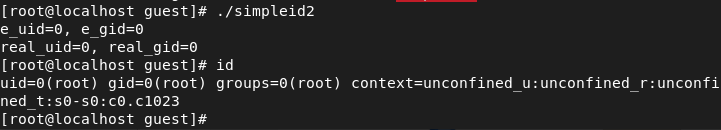


Команда chown root:guest /home/guest/simpleid2 меняет владельца файла. Команда  
chmod u+s /home/guest/simpleid2 меняет права доступа к файлу.

Проверю правильность установки новых атрибутов и смены владельца файла  
simpleid2 командой: ls -l simpleid2



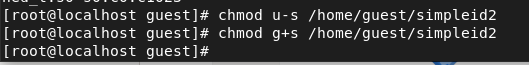
Запущу simpleid2 и id, команды: ./simpleid2 и id



После выполнения команд изменился параметр e\_uid.

##### SetGID-бит

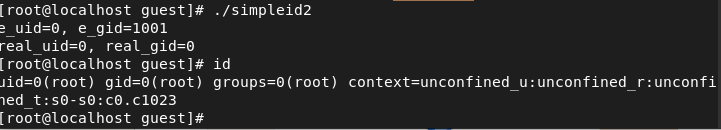
От имени суперпользователя выполню команды:  
chmod u-s /home/guest/simpleid2 – чтобы отменить изменения на прошлом шаге  
chmod g+s /home/guest/simpleid2



Проверю правильность установки новых атрибутов и смены владельца файла  
simpleid2 командой: ls -l simpleid2



Запущу simpleid2 и id, команды: ./simpleid2 и id. Ничего не изменилось.



Создам программу readfile.c



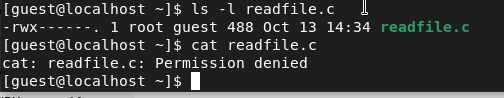
Скомпилирую её командой: gcc readfile.c -o readfile



Сменю владельца у файла readfile.c и изменю права так, чтобы только  
суперпользователь (root) мог прочитать его, a guest не мог.



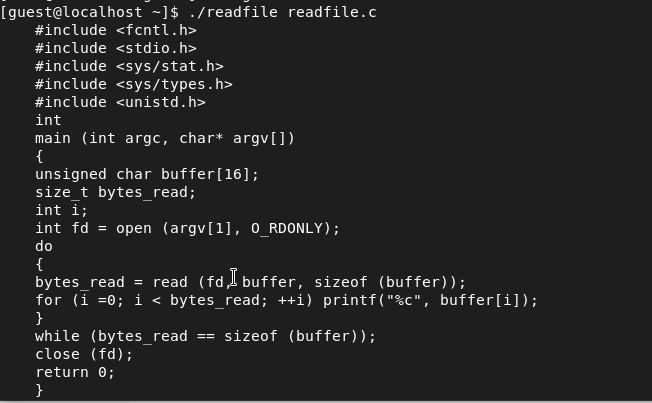
Проверю, что пользователь guest не может прочитать файл readfile.c.



Сменю у программы readfile владельца и установлю SetUID-бит.



Проверю, может ли программа readfile прочитать файл readfile.c



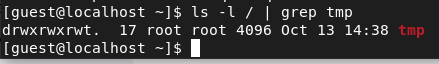
Проверю, может ли программа readfile прочитать файл /etc/shadow



Поскольку у программы установлен SetUID-бит, то ей временно предоставляются  
права владельца файла (суперпользователя). Поэтому программа может прочитать  
файл с правами доступа только для владельца суперпользователя.

##### Исследование Sticky-бита

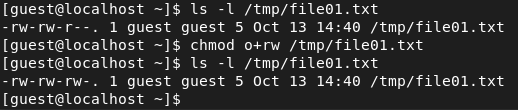
Выясню, установлен ли атрибут Sticky на директории /tmp, для чего выполню  
команду ls -l / | grep tmp



От имени пользователя guest создам файл file01.txt в директории /tmp со словом test:  
echo "test" > /tmp/file01.txt



Просмотрю атрибуты у только что созданного файла и разрешу чтение и запись для  
категории пользователей «все остальные»:  
ls -l /tmp/file01.txt  
chmod o+rw /tmp/file01.txt  
ls -l /tmp/file01.txt



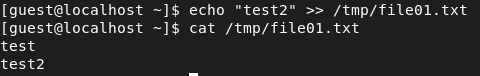
От пользователя guest2 (не являющегося владельцем) попробую прочитать файл  
/tmp/file01.txt: cat /tmp/file01.txt



От пользователя guest2 попробую дозаписать в файл /tmp/file01.txt слово test2  
командой echo "test2" >> /tmp/file01.txt

Мне удалось выполнить операцию.

Проверю содержимое файла командой cat /tmp/file01.txt



От пользователя guest2 попробую записать в файл /tmp/file01.txt слово test3, стерев  
при этом всю имеющуюся в файле информацию командой echo "test3" >  
/tmp/file01.txt

Мне удалось выполнить операцию.

Проверю содержимое файла командой cat /tmp/file01.txt



От пользователя guest2 попробую удалить файл /tmp/file01.txt командой rm  
/tmp/file01.txt

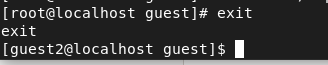


Мне не удалось удалить файл.

Повышу свои права до суперпользователя следующей командой su и выполню после  
этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp



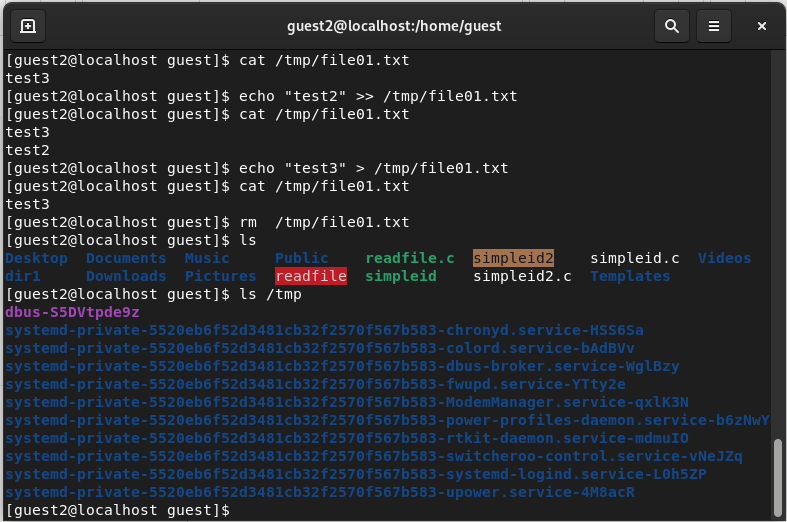
Покину режим суперпользователя командой exit



От пользователя guest2 проверьте, что атрибута t у директории /tmp нет: ls -l / | grep  
tmp



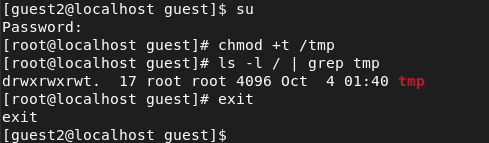
Повторю предыдущие шаги.



Мне удалось удалить файл от имени пользователя, не являющегося его владельцем.  
Это связано с тем, что Sticky-bit позволяет защищать файлы от случайного удаления,  
когда несколько пользователей имеют права на запись в один и тот же каталог. Если

у файла атрибут t стоит, значит пользователь может удалить файл, только если он  
является пользователем-владельцем файла или каталога, в котором содержится файл.  
Если же этот атрибут не установлен, то удалить файл могут все пользователи,  
которым позволено удалять файлы из каталога.

Повышу свои права до суперпользователя и верну атрибут t на директорию /tmp:  
su  
chmod +t /tmp  
exit



### Вывод

В ходе данной лабораторной работы я изучила механизмы изменения  
идентификаторов, применения SetUID-, SetGID- и Sticky-битов. Рассмотрела работу  
механизма смены идентификатора процессов пользователей, а также влияние бита  
Sticky на запись и удаление файлов.

# Список литературы

* [Кулябов Д. С., Королькова А. В., Геворкян М. Н Лабораторная работа №5](https://esystem.rudn.ru/pluginfile.php/1651751/mod_resource/content/3/004-lab_discret_extattr.pdf)