РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

**Факультет физико-математических и естественных наук Кафедра прикладной информатики и теории вероятностей**

**ЛАБОРАТОРНОЙ РАБОТЕ № 5**

*дисциплина: Информационная безопасность*

Студент: Нгуен Фыок Дат Группа: НФИбд-01-20

# МОСКВА

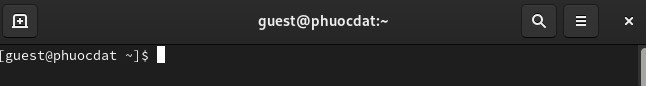
2023 г

**Постановка задачи** Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными

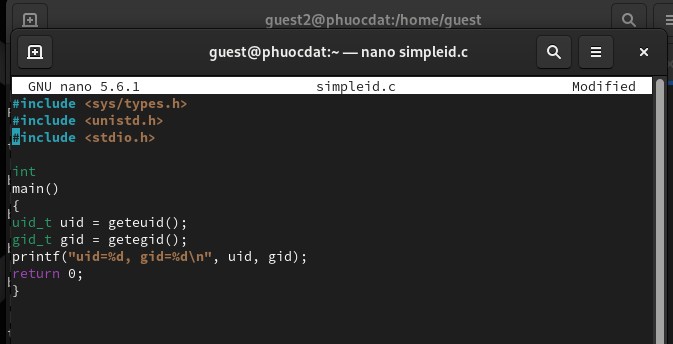
атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

**Выполнение работы** Создание программы

1. Войду в систему от имени пользователя guest



1. Создам программу simpleid.c



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



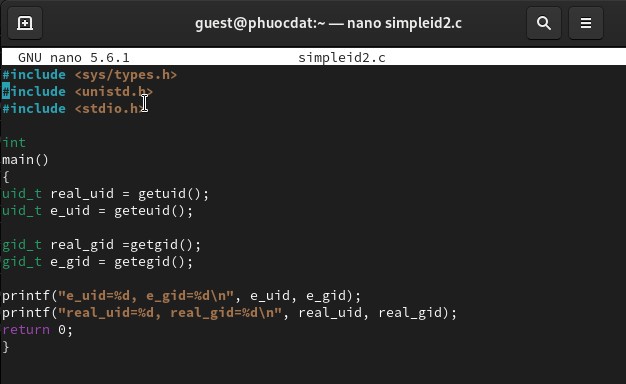
1. Выполню программу simpleid командой ./simpleid



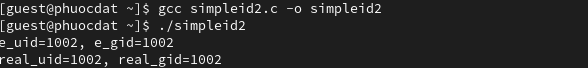
1. Выполню системную программу id командой id. Результат совпадает



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



1. Скомпилирую и запущу simpleid2.c



1. Работа с е SetUID-битом
2. 1. От имени суперпользователя выполню команды: chown root:guest

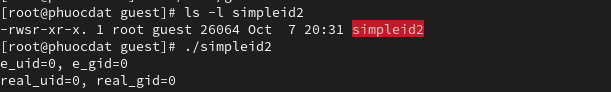
/home/guest/simpleid2 chmod u+s /home/guest/simpleid2



* 1. Команда chown root:guest /home/guest/simpleid2 меняет владельца файла. Команда chmod u+s /home/guest/simpleid2 меняет права доступа к файлу.
  2. Проверю правильность установки новых атрибутов и смены владельца файла simpleid2 командой: ls -l simpleid2



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



После выполнения команд изменился параметр e\_uid.

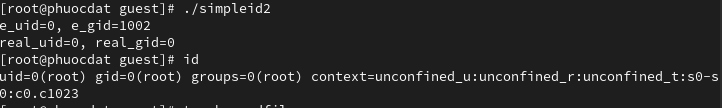
1. Проделаю тоже самое относительно SetGID-бита
2. 1 От имени суперпользователя выполню команды: chmod u-s /home/guest/simpleid2 – чтобы отменить изменения на прошлом шаге chmod g+s /home/guest/simpleid2



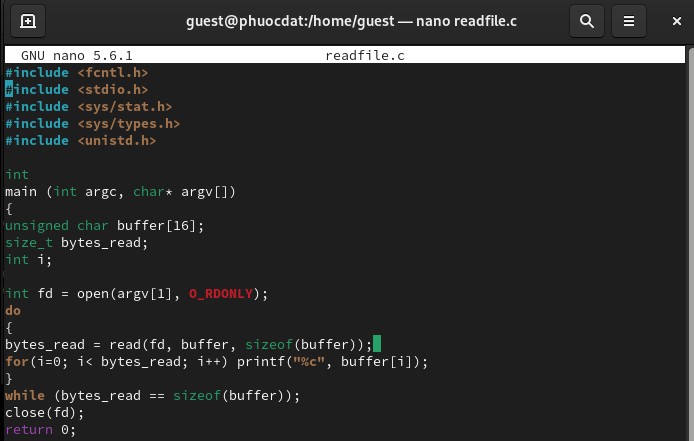
* 1. Проверю правильность установки новых атрибутов и смены владельца файла simpleid2 командой: ls -l simpleid2



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



1. Создам программу readfile.c



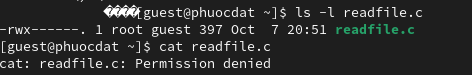
1. Скомпилирую её командой: gcc readfile.c -o readfile



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



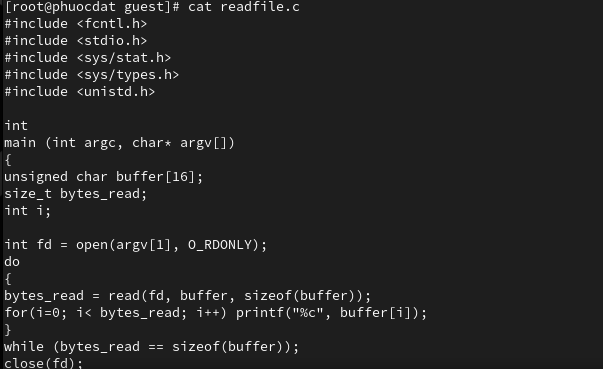
1. Проверю, что пользователь guest не может прочитать файл readfile.c.



1. Сменю у программы readfile владельца и установлю SetUID-бит.



1. Проверю, может ли программа readfile прочитать файл readfile.c



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



1. 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

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

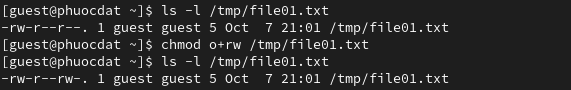


1. От имени пользователя guest создам файл file01.txt в директории /tmp со словом test: echo "test" > /tmp/file01.txt



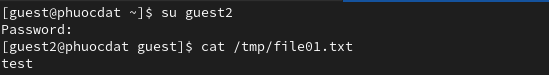
1. Просмотрю атрибуты у только что созданного файла и разрешу чтение и запись для категории пользователей «все остальные»: ls -l /tmp/file01.txt chmod o+rw

/tmp/file01.txt ls -l /tmp/file01.txt



1. От пользователя guest2 (не являющегося владельцем) попробую прочитать файл

/tmp/file01.txt: cat /tmp/file01.txt



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



Мне удалось выполнить операцию. 6. Проверю содержимое файла командой cat

/tmp/file01.txt



1. 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.

1. Проверю содержимое файла командой cat /tmp/file01.txt



Мне не удалось удалить файл. 10. Повышу свои права до суперпользователя следующей командой su и выполню после этого команду, снимающую атрибут t (Sticky-бит) с

директории /tmp: chmod -t /tmp



1. Покину режим суперпользователя командой exit

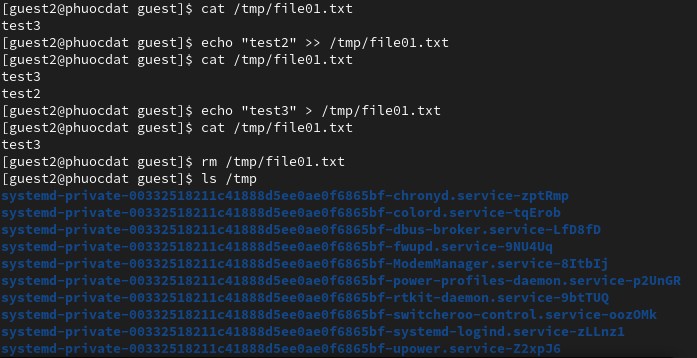


1. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет: ls -l /

| grep tmp



1. Повторю предыдущие шаги.

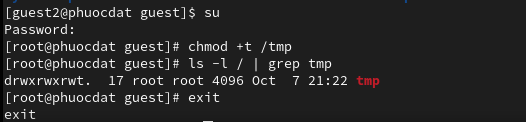


1. Мне удалось удалить файл от имени пользователя, не являющегося его владельцем.

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.

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



**Заключение** В ходе данной лабораторной работы я изучила механизмы изменения

идентификаторов, применения SetUID-, SetGID- и Sticky-битов. Рассмотрела работ

механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.