**Федеральное государственное автономное образовательное учреждение высшего образования**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»**

**(РУДН)**

Факультет физико-математических и естественных наук

Кафедра информационных технологий

**ОТЧЕТ  
по лабораторной работе 5**

ТЕМА «**Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов**»  
по дисциплине «Информационная безопасность»

Выполнил:

Студент группы НПИбд-02-21

Студенческий билет № 1032205641

Сатлихана Петрити

(Подпись)

Москва 2024

# Список содержания

[Список содержания 2](#_Toc178585426)

[Список изображений 3](#_Toc178585427)

[Цель работы 4](#_Toc178585428)

[*Последовательность выполнения работы* 4](#_Toc178585429)

[Выводы 9](#_Toc178585430)

# Список изображений

[рис. 1 Создание программы simpleid.c 6](#_Toc178584846)

[рис. 2 Пункты 3,4,5: компиляция и id 6](#_Toc178584847)

[рис. 3 Создание программы simpleid.c(вместо simpleid2.c) 7](#_Toc178584848)

[рис. 4 Компиляция и запуск simpleid2.c 7](#_Toc178584849)

[рис. 5 выполнение команд chown,chmod 7](#_Toc178584850)

[рис. 6 проверка правильности установки новых атрибутов 8](#_Toc178584851)

[рис. 7 запуск и id 8](#_Toc178584852)

[рис. 8 SetGID-бита. 8](#_Toc178584853)

[рис. 9 Создание программы readfile.c 8](#_Toc178584854)

[рис. 10 Пункты 14,15,16 9](#_Toc178584855)

[рис. 11 Пункты 17,18,19 9](#_Toc178584856)

[рис. 12 Пункты 1,2,3 10](#_Toc178584857)

[рис. 13 Пункты 4,5,6,7,8 10](#_Toc178584858)

[рис. 14 Пункты 10,11 11](#_Toc178584859)

[рис. 15 Hет aтрибута t у директории /tmp 11](#_Toc178584860)

[рис. 16 Возвращающий атрибут t 11](#_Toc178584861)

[рис. 17 Проверка 11](#_Toc178584862)

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.

*Последовательность выполнения работы*

**5.3. Порядок выполнения работы**

**5.3.1. Создание программы**

1. Войдите в систему от имени пользователя guest.

2. Создайте программу simpleid.c:

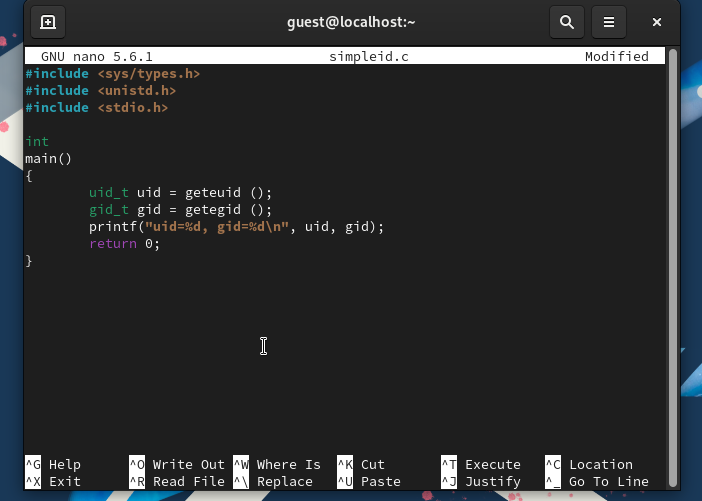


рис. 1 Создание программы simpleid.c

3. Скомплилируйте программу и убедитесь, что файл программы создан:

gcc simpleid.c -o simpleid

4. Выполните программу simpleid:

./simpleid

5. Выполните системную программу id:

id и сравните полученный вами результат с данными предыдущего пункта задания.  
Один и тот же uid и gid

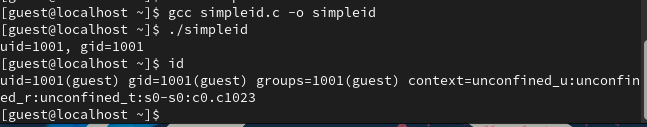


рис. 2 Пункты 3,4,5: компиляция и id

6. Усложните программу, добавив вывод действительных идентификаторов:

Получившуюся программу назовите simpleid2.c.



рис. 3 Создание программы simpleid.c(вместо simpleid2.c)

7. Скомпилируйте и запустите simpleid2.c:

gcc simpleid2.c -o simpleid2

./simpleid2

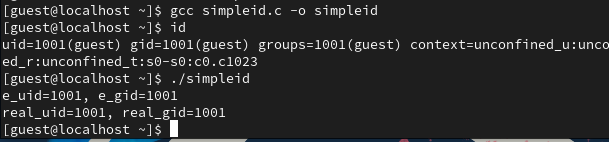
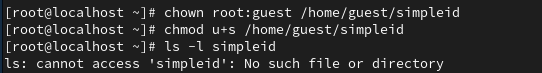


рис. 4 Компиляция и запуск simpleid2.c

8. От имени суперпользователя выполните команды:

chown root:guest /home/guest/simpleid2

chmod u+s /home/guest/simpleid2

  
 рис. 5 выполнение команд chown,chmod

9. Используйте sudo или повысьте временно свои права с помощью su.

Поясните, что делают эти команды.

**sudo (SuperUser DO)**

* **Что это?** sudo — это команда, которая позволяет пользователям выполнять команды с привилегиями суперпользователя (root) или другого пользователя, указанного в конфигурационном файле /etc/sudoers.

**su (Substitute User)**

* **Что это?** su — это команда, которая позволяет пользователю переключиться на другого пользователя, обычно на суперпользователя (root).

10. Выполните проверку правильности установки новых атрибутов и смены

владельца файла simpleid2:

ls -l simpleid2

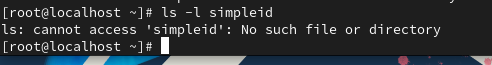


рис. 6 проверка правильности установки новых атрибутов

11. Запустите simpleid2 и id:

./simpleid2

id Сравните результаты.

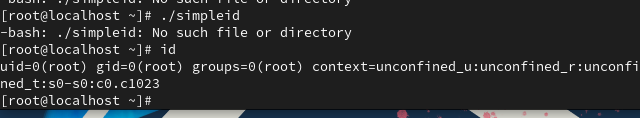


рис. 7 запуск и id

12. Проделайте тоже самое относительно SetGID-бита.

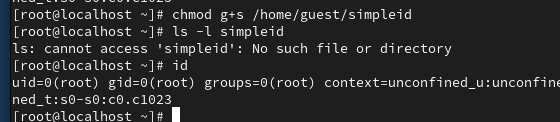


рис. 8 SetGID-бита.

13. Создайте программу readfile.c:

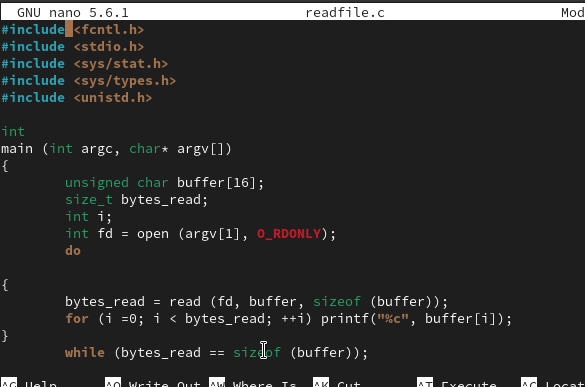


рис. 9 Создание программы readfile.c

14. Откомпилируйте её.

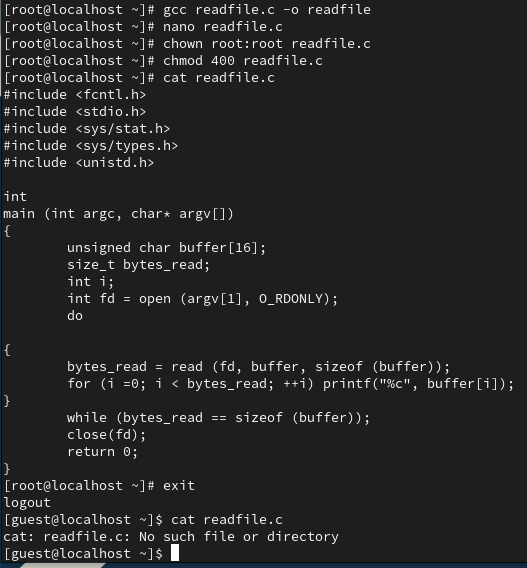
gcc readfile.c -o readfile

15. Смените владельца у файла readfile.c (или любого другого текстового

файла в системе) и измените права так, чтобы только суперпользователь

(root) мог прочитать его, a guest не мог.

16. Проверьте, что пользователь guest не может прочитать файл readfile.c.

  
 рис. 10 Пункты 14,15,16

17. Смените у программы readfile владельца и установите SetU’D-бит.

18. Проверьте, может ли программа readfile прочитать файл readfile.c?

19. Проверьте, может ли программа readfile прочитать файл /etc/shadow? Да, прочитается.

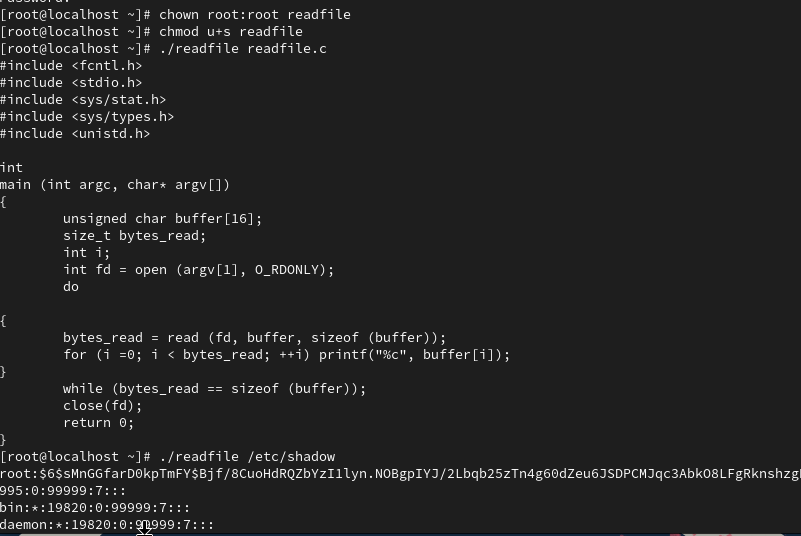


рис. 11 Пункты 17,18,19

**5.3.2. Исследование Sticky-бита**

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего

выполните команду

ls -l / | grep tmp

2. От имени пользователя guest создайте файл file01.txt в директории /tmp

со словом test:

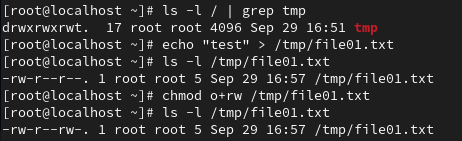
echo "test" > /tmp/file01.txt

3. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»:

ls -l /tmp/file01.txt

chmod o+rw /tmp/file01.txt

ls -l /tmp/file01.txt

  
 рис. Пункты 1,2,3

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

cat /tmp/file01.txt

5. От пользователя guest2 попробуйте дозаписать в файл

/tmp/file01.txt слово test2 командой echo "test2" > /tmp/file01.txt Удалось ли вам выполнить операцию?

6. Проверьте содержимое файла командой

cat /tmp/file01.txt

7. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt

слово test3, стерев при этом всю имеющуюся в файле информацию командой

echo "test3" > /tmp/file01.txt

Удалось ли вам выполнить операцию?

8. Проверьте содержимое файла командой

cat /tmp/file01.txt

9. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой

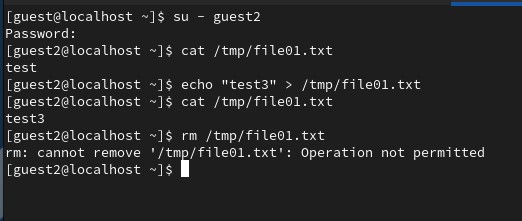
rm /tmp/fileOl.txt Удалось ли вам удалить файл? Нет, операция была запрещена.

рис. 13 Пункты 4,5,6,7,8

10. Повысьте свои права до суперпользователя следующей командой

su - и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp:

chmod -t /tmp

11. Покиньте режим суперпользователя командой

Exit

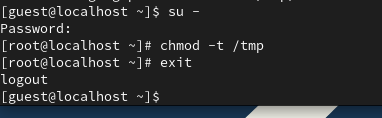
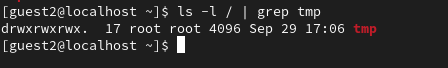


рис. Пункты 10,11

12. От пользователя guest2 проверьте, что атрибута t у директории /tmp

нет:

ls -l / | grep tmp



*рис. 15 Hет aтрибута t у директории /tmp*

13. Повторите предыдущие шаги. Какие наблюдаются изменения?

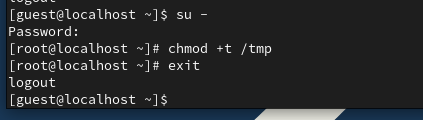
14. Удалось ли вам удалить файл от имени пользователя, не являющегося

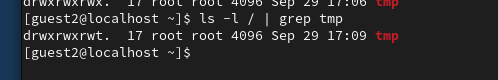
его владельцем? Ваши наблюдения занесите в отчёт.

15. Повысьте свои права до суперпользователя и верните атрибут t на директорию /tmp:

su - chmod +t /tmp

exit

 рис. Возвращающий атрибут t



*рис. 17 Проверка*

Выводы

В этой лабораторной работе мы изучили, как работают специальные атрибуты файлов в Linux, такие как SetUID, SetGID и Sticky-бит. Мы увидели, как они помогают контролировать доступ к файлам и программам, а также защищают общие папки от удаления файлов другими пользователями. Эти механизмы важны для безопасности и защиты данных в системах с несколькими пользователями.