Отчёт по лабораторной работе № 5

Информационная безопасность

Кинсиклунон Доря Флора

Содержание

## 0.1 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.

## 0.2 Теоретическое введение

SetUID, SetGID и Sticky - это специальные типы разрешений позволяют задавать расширенные права доступа на файлы или каталоги. • SetUID (set user ID upon execution — «установка ID пользователя во время выполнения) являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами владельца исполняемого файла. • SetGID (set group ID upon execution — «установка ID группы во время выполнения») являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами группы исполняемого файла. • Sticky bit в основном используется в общих каталогах, таких как /var или /tmp, поскольку пользователи могут создавать файлы, читать и выполнять их, принадлежащие другим пользователям, но не могут удалять файлы, принадлежащие другим пользователям.

## 0.3 Выполнение лабораторной работы

## 0.4 5.1 Создание программы

Для начала я убедилась, что компилятор gcc установлен, исолпьзуя команду “gcc -v”. Затем отключила систему запретов до очередной перезагрузки системы командой “sudo setenforce 0”, после чего команда “getenforce” вывела “Permissive” (рис. 5.1).

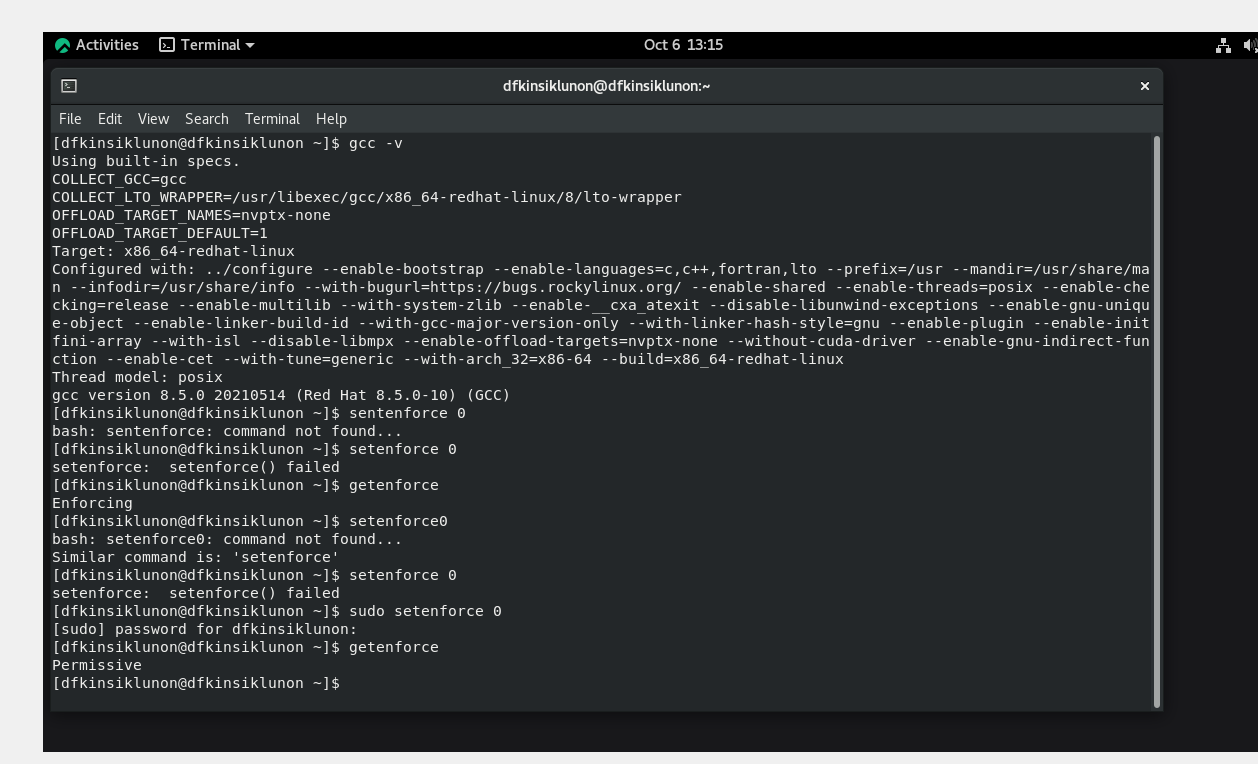


Рис. 5.1: Предварительная подготовка

Проверила успешное выполнение команд “whereis gcc” и “whereis g++” (их расположение) (рис. 5.2).

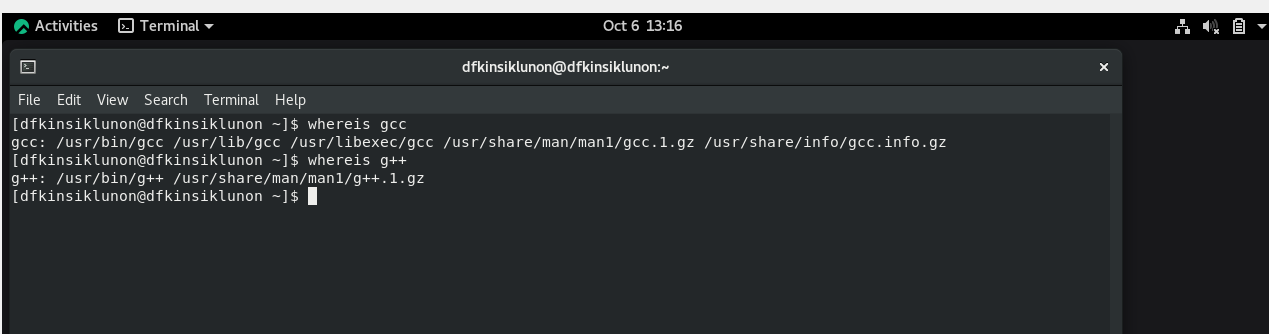


Рис. 5.2: Команда “whereis”

Вошла в систему от имени пользователя guest командой “su - guest”. Создала программу simpleid.c командой “touch simpleid.c” и открыла её в редакторе командой “gedit /home/guest/simpleid.c” (рис. 5.3).

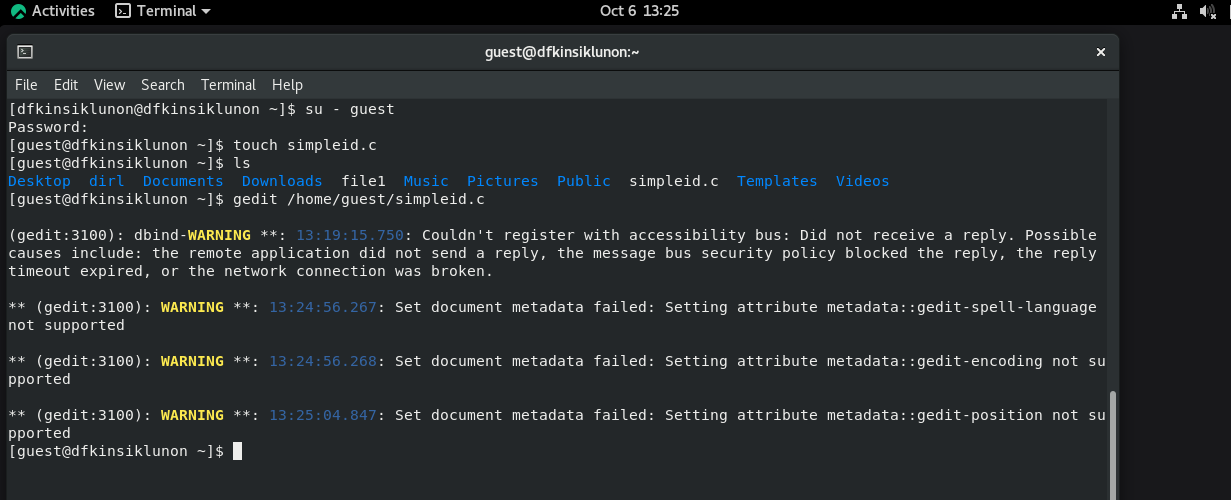


Рис. 5.3: Вход в систему и создание программы

Код программы выглядит следующим образом (рис. 5.4).

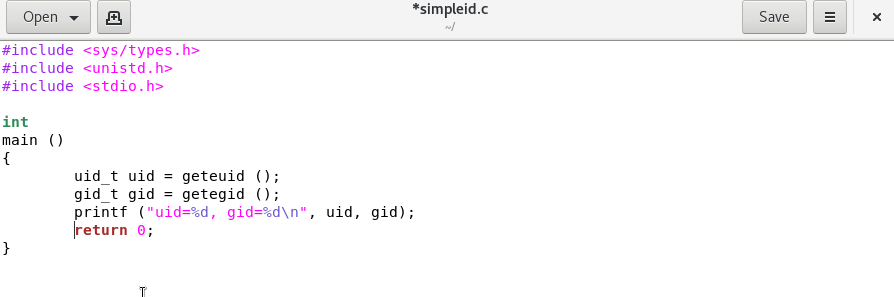


Рис. 5.4: Код программы simpleid.c

Скомпилировала программу и убедилась, что файл программы был создан командой “gcc simpleid.c -o simpleid”. Выполнила программу simpleid командой “./simpleid”, а затем выполнила системную программу id командой “id”. Результаты, полученные в результате выполнения обеих команд, совпадают (uid=1001 и gid=1001) (рис. 5.5).



Рис. 5.5: Компиляция и выполнение программы simpleid

Усложнила программу, добавив вывод действительных идентификаторов (рис. 5.6).

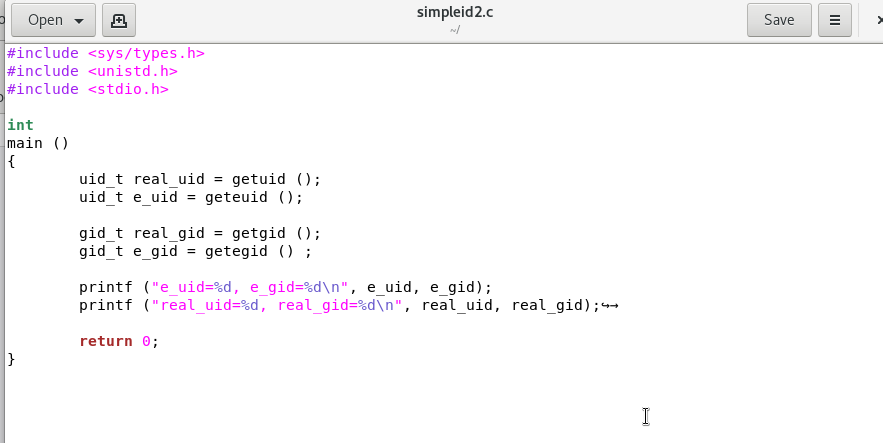


Рис. 5.6: Усложнение программы

Получившуюся программу назвала simpleid2.c (рис. 3.7).



Рис. 5.7: Переименование программы в simpleid2.c

Скомпилировала и запустила simpleid2.c командами “gcc simpleid2.c -o sipleid2” и “./simpleid2” (рис. 3.8).

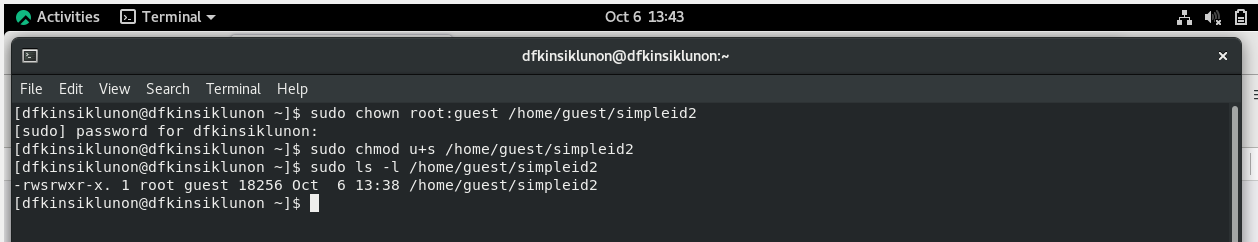


Рис. 5.8: Компиляция и выполнение программы simpleid2

От имени суперпользователя выполнила команды “sudo chown root:guest /home/guest/simpleid2” и “sudo chmod u+s /home/guest/simpleid2”, затем выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой “sudo ls -l /home/guest/simpleid2” (рис. 3.9). Этими командами была произведена смена пользователя файла на root и установлен SetUID-бит.

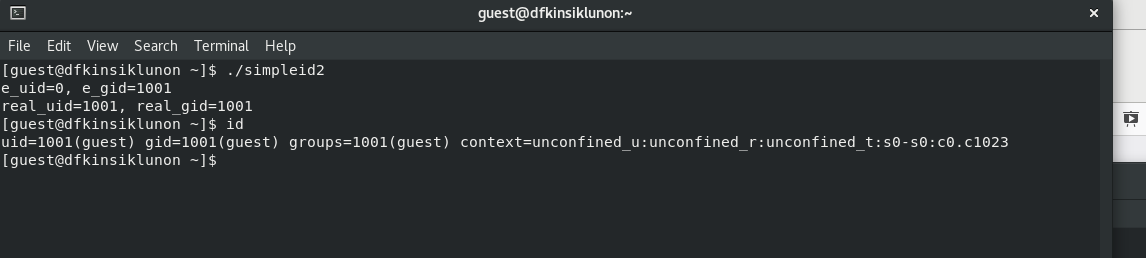


Рис. 5.9: Установка новых атрибутов (SetUID) и смена владельца файла

Запустила программы simpleid2 и id. Теперь появились различия в uid (рис. 5.10).



Рис. 5.10: Запуск simpleid2 после установки SetUID

Проделала тоже самое относительно SetGID-бита. Также можем заметить различия с предыдущим пунктом (рис. 5.11).

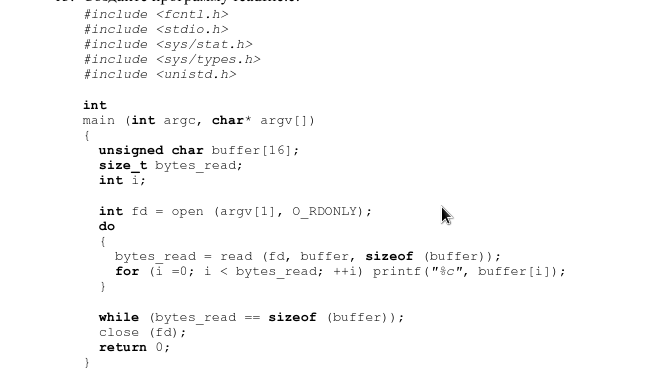


Рис. 5.10: Запуск simpleid2 после установки SetUID

Создаем программу readfile.c (рис. 5.12).

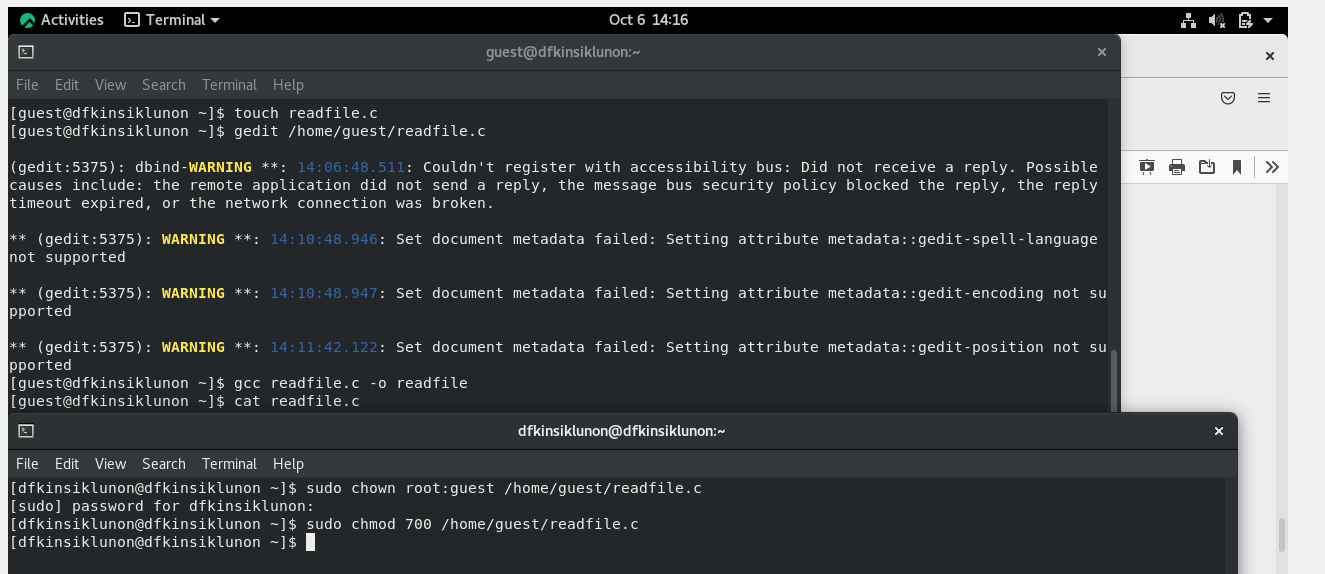


Рис. 5.12: Код программы readfile.c

Скомпилировала созданную программу командой “gcc readfile.c -o readfile”. Сменила владельца у файла readfile.c командой “sudo chown root:guest /home/guest/readfile.c” и поменяла права так, чтобы только суперпользователь мог прочитать его, а guest не мог, с помощью команды “sudo chmod 700 /home/guest/readfile.c”. Теперь убедилась, что пользователь guest не может прочитать файл readfile.c командой “cat readfile.c”, получив отказ в доступе (рис. 5.13).

|  |
| --- |
| Рис. 5.13: Смена владельца и прав доступа у файла readfile.c |

Рис. 5.13: 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. 5.14).

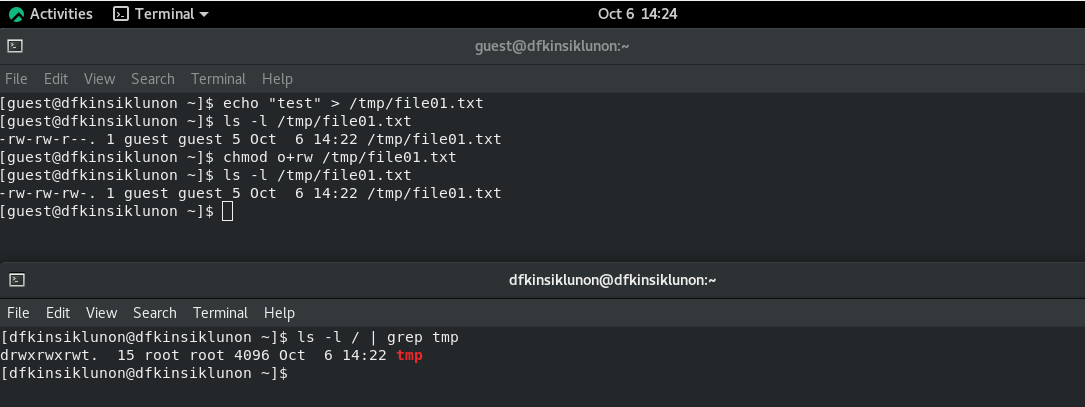


Рис. 5.14: Запуск программы readfile

## 0.5 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. 5.15).

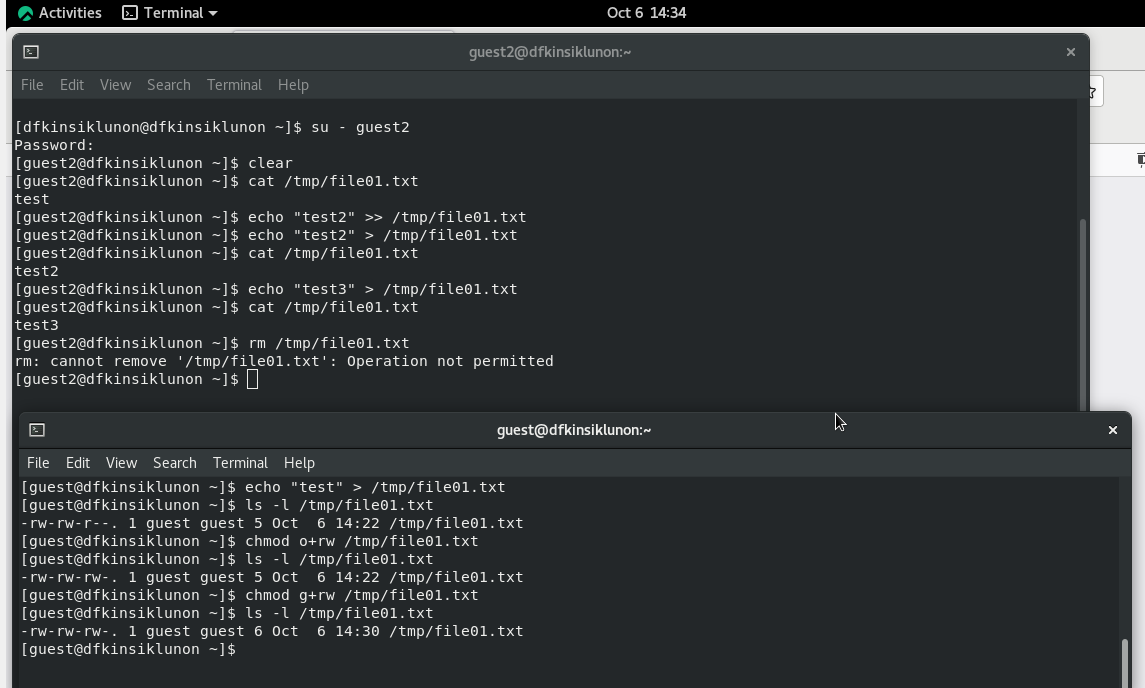


Рис. 5.15: Создание файла file01.txt

От имени пользователя guest2 попробовала прочитать файл командой “cat /tmp/file01.txt” - это удалось. Далее попыталась 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 попробовала удалить файл - это не удается ни в каком из случаев, возникает ошибка (рис. 5.16).

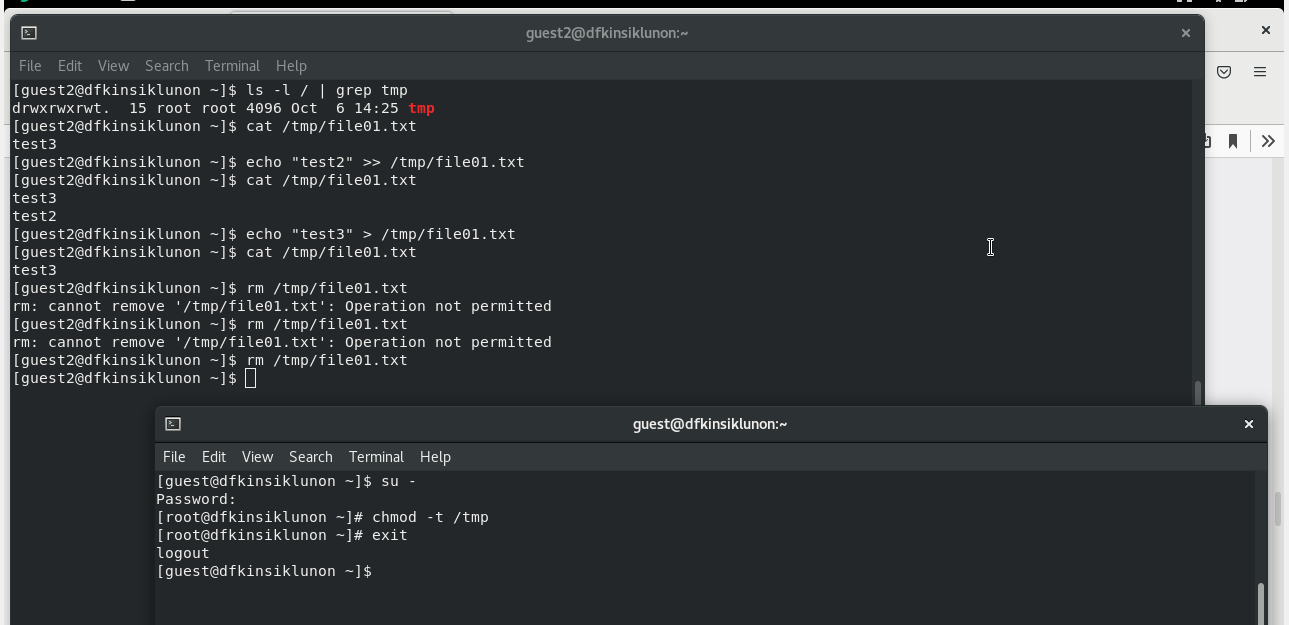


Рис. 5.16: Попытка выполнить действия над файлом file01.txt от имени пользователя guest2

Повысила права до суперпользователя командой “su -” и выполнила команду, снимающую атрибут t с директории /tmp “chmod -t /tmp”. После чего покинула режим суперпользователя командой “exit”. Повторила предыдущие шаги. Теперь мне удалось удалить файл file01.txt от имени пользователя, не являющегося его владельцем (рис. 5.17).

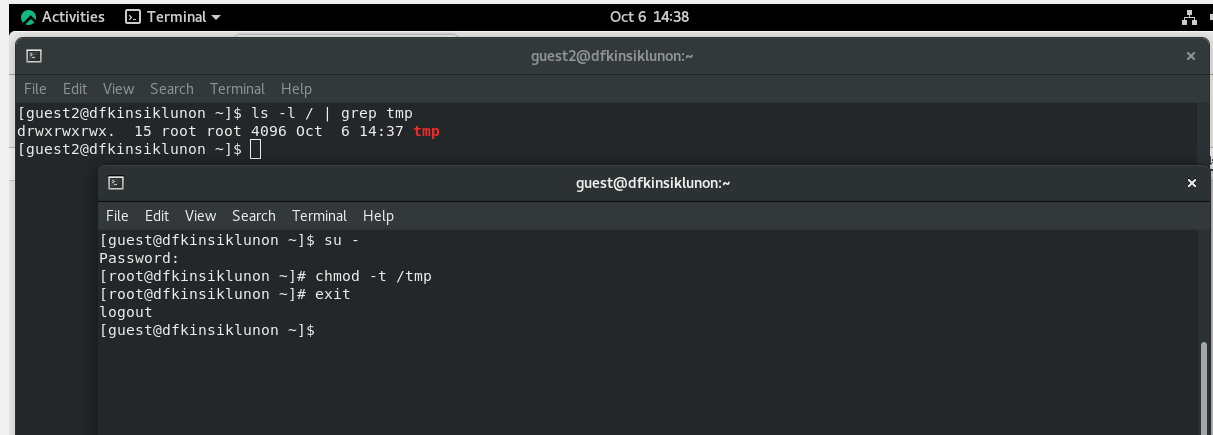


Рис. 5.17: Удаление атрибута t (Sticky-бита) и повторение действий

Повысила свои права до суперпользователя и вернула атрибут t на директорию /tmp (рис. 5.18).

|  |
| --- |
| Рис. 5.18: Возвращение атрибута t (Sticky-бита) |

Рис. 5.18: Возвращение атрибута t (Sticky-бита)

## 0.6 Выводы

ходе выполнения данной лабораторной работы я изучила механизмы изменения идентификаторов, применение SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.