Отчёт по лабораторной работе №5

Дисциплина: Основы информационной безопасности

Мишина Анастасия Алексеевна

Содержание

# 1 Цель работы

Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в кон- соли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов [**infosec?**].

# 2 Выполнение лабораторной работы

## 2.1 Создание программы

Входим в систему от имени пользователя guest и создаем файл simpleid.c , открываем его в редакторе vi (рис. 1). В файл вписываем код из мануала (рис. 2).

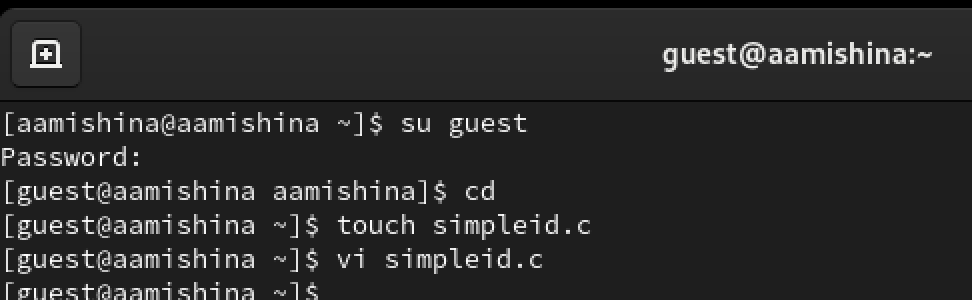


Рис. 1: Создание файла simpleid.c

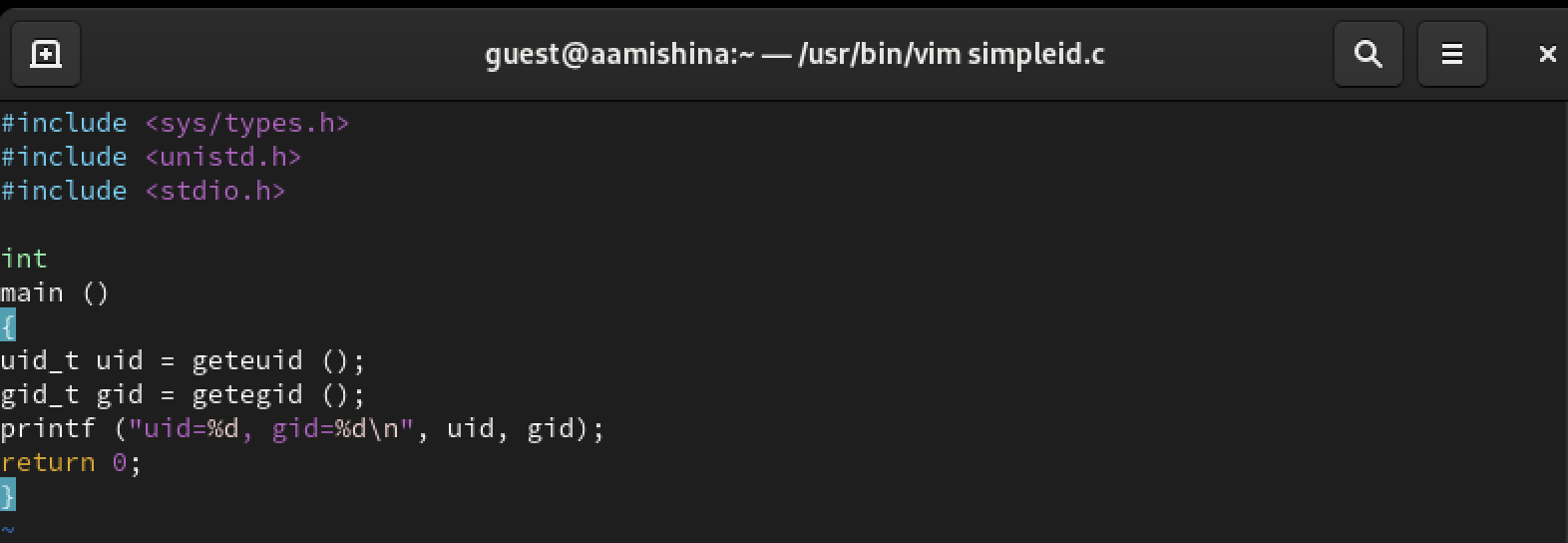


Рис. 2: Программа в файле simpleid.c

Скомплилируем программу и убедимся, что файл программы создан: gcc simpleid.c -o simpleid. Выполняем программу simpleid: ./simpleid и выполняем системную программу id: id. Видим, что результаты совпадают (рис. 3).

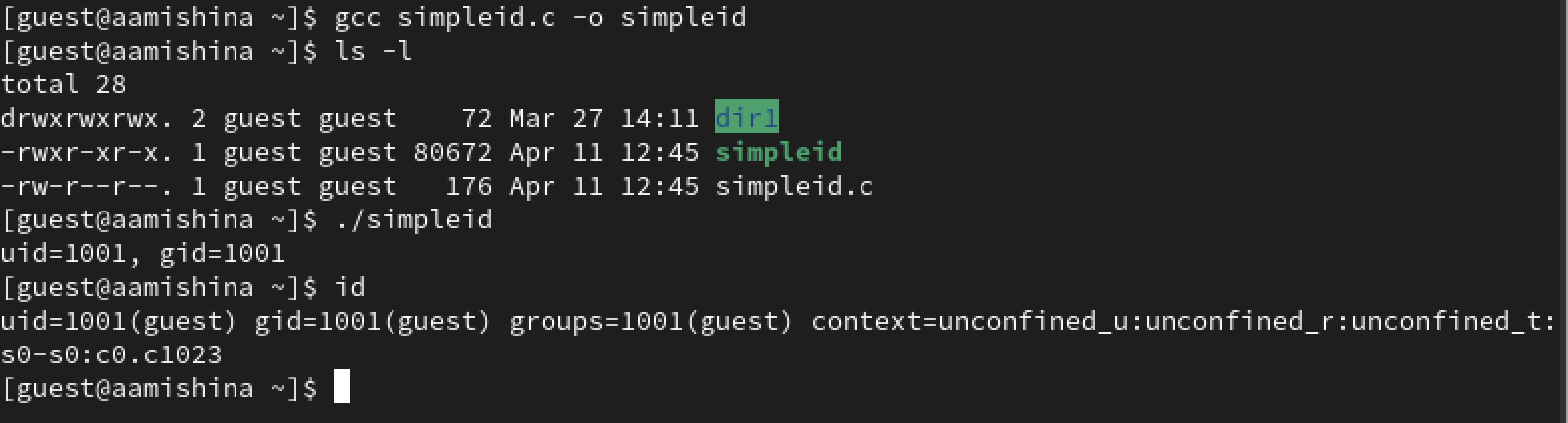


Рис. 3: Выполнение программы simpleid и команды id

Теперь усложним программу, добавив вывод действительных идентификаторов (рис. 4).



Рис. 4: Добавление вывода действительных идентификаторов

Компилируем и запускаем simpleid2.c: gcc simpleid2.c -o simpleid2 и ./simpleid2 (рис. 5).

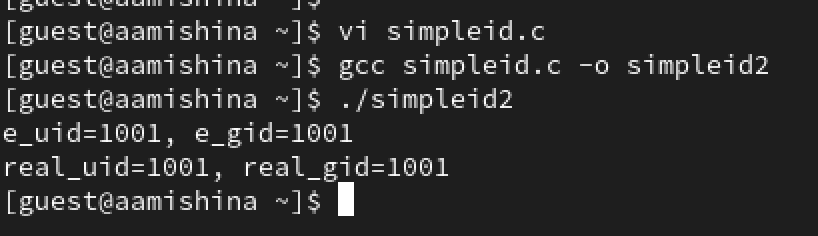


Рис. 5: Компиляция и запуск simpleid2

Переходим в режим суперпользователя и выполняем команды: chown root:guest /home/guest/simpleid2 и chmod u+s /home/guest/simpleid2. От имени суперпользователя мы изменили владельца файла и добавили атрибут s, это означает, что пользователь будет выполнять файл с разрешениями владельца файла. Проверяем правильность установки: ls -l simpleid2. Запусакем simpleid2 и вводим команду id: ./simpleid2 и id. Теперь владельцем файла является пользователь с id 0 (root), а изначально владельцем был пользователь с id 1001 (guest) (рис. 6).

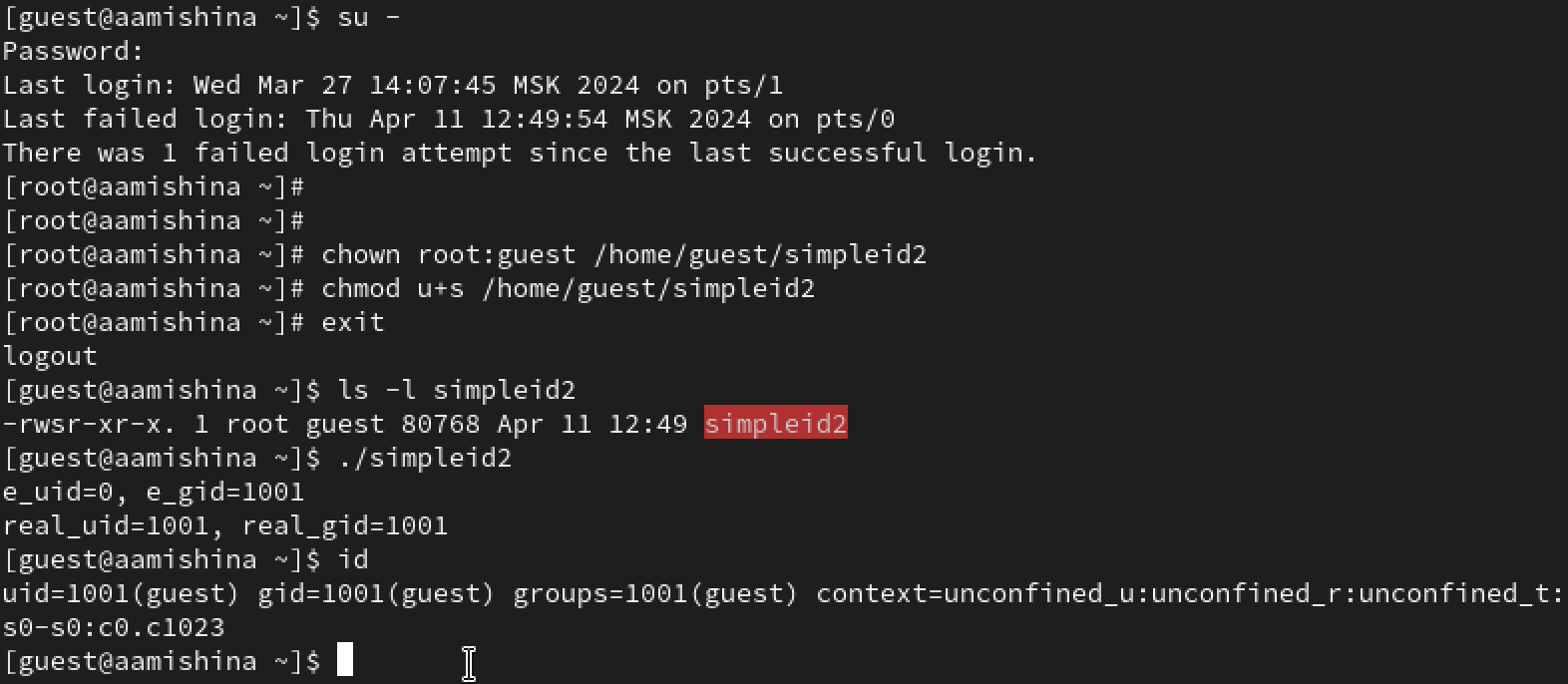


Рис. 6: Запуск simpleid2 с SETUID. Сравнение результатов

Повторяем тоже самое относительно SetGID-бита (рис. 7).

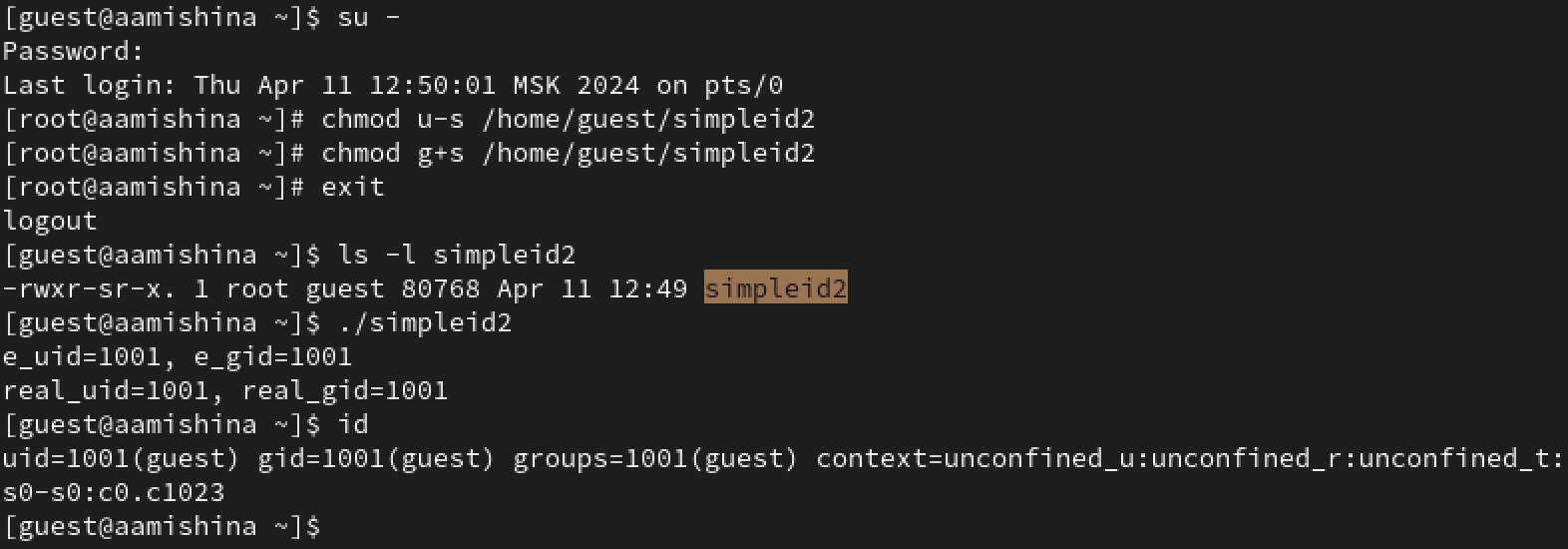


Рис. 7: Запуск simpleid2 с SETGID. Сравнение результатов

Создаем программу readfile.c (рис. 8).



Рис. 8: Программа readfile

Компилируем её: gcc readfile.c -o readfile. Меняем владельца у файла readfile.c и изменяем права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог. Проверяем, что пользователь guest не может прочитать файл readfile.c (рис. 9).

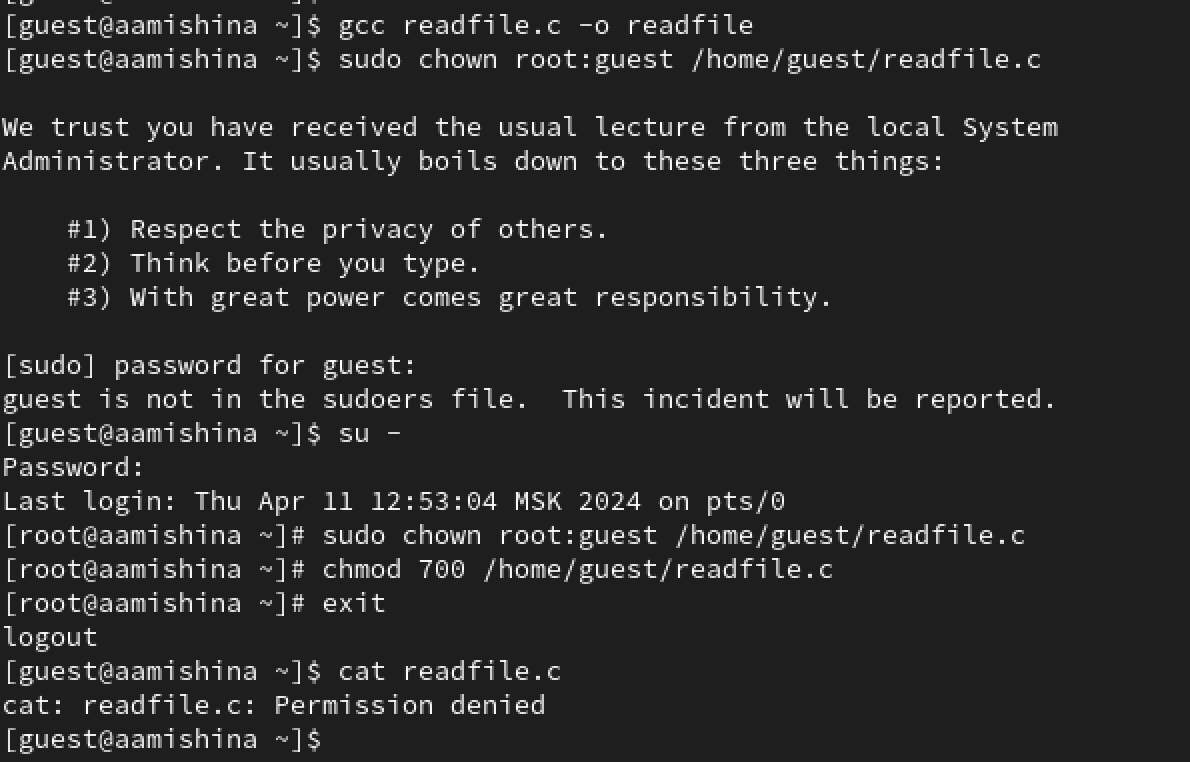


Рис. 9: Изменение прав доступа, проверка от имени пользователя guest

Меняем у программы readfile владельца и становим SetUID-бит. Проверяем, может ли программа readfile прочитать файл readfile.c (может). Проверяем, может ли программа readfile прочитать файл /etc/shadow (может) (рис. 10).

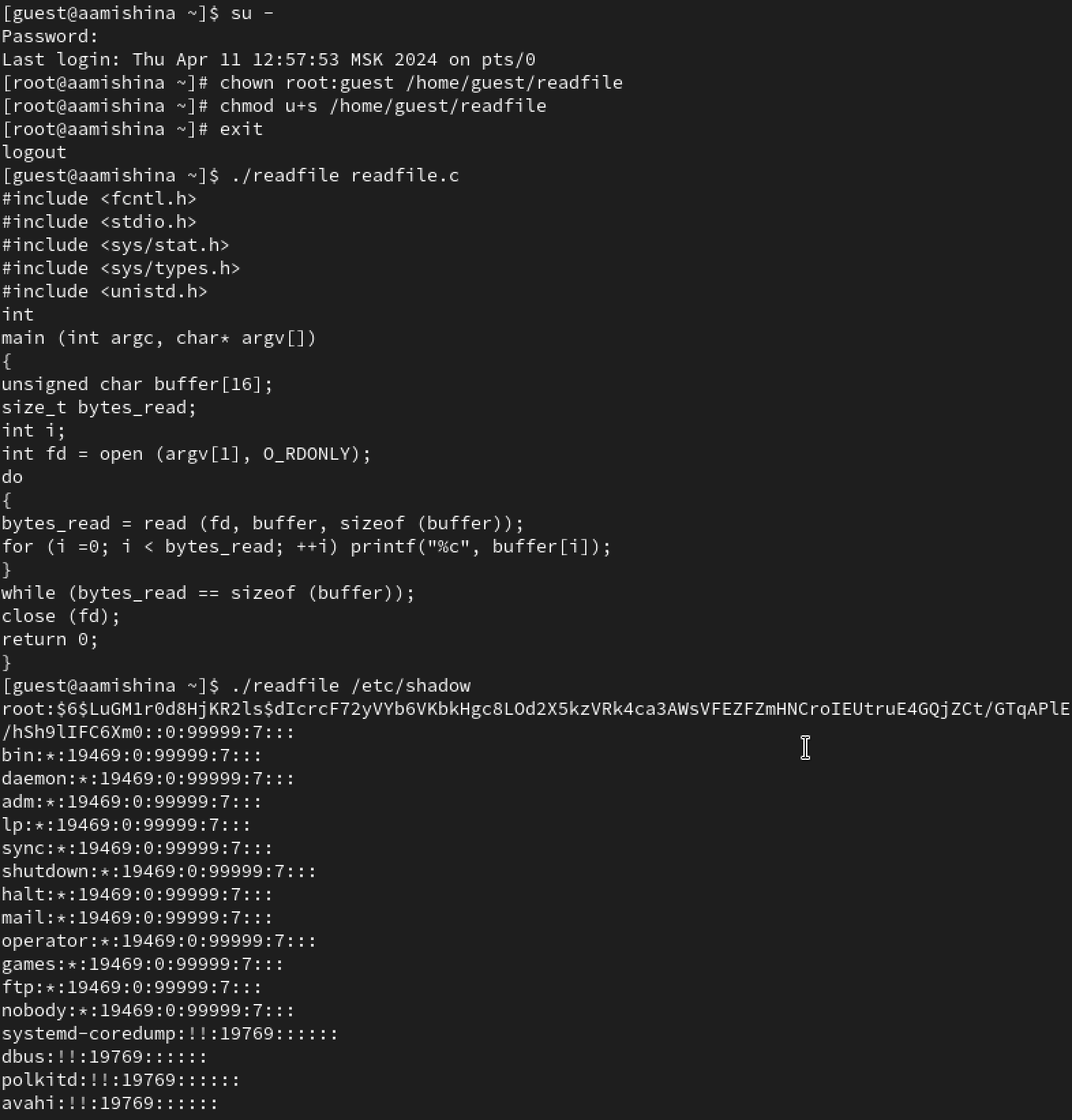


Рис. 10: Установка SETUID для readfile, проверка

## 2.2 Исследование Sticky-бита

Выясним, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду ls -l / | grep tmp. От имени пользователя guest создаем файл file01.txt в директории /tmp со словом test: echo “test” > /tmp/file01.txt. Просматриваем атрибуты у только что созданного файла и разрешаем чтение и запись для категории пользователей «все остальные»: ls -l /tmp/file01.txt, chmod o+rw /tmp/file01.txt и ls -l /tmp/file01.txt. От пользователя guest2 (не являющегося владельцем) пробуем прочитать файл /tmp/file01.txt: cat /tmp/file01.txt (успешно), дозаписать в файл слово test2 командой echo “test2” > /tmp/file01.txt (отказано в доступе), записать в файл слово test3, стерев при этом всю имеющуюся в файле информацию: echo “test3” > /tmp/file01.txt (отказано в доступе). Проверяем содержимое файла: cat /tmp/file01.txt (не изменилось). Попробуем удалить файл: rm /tmp/fileOl.txt (отказано в доступе) (рис. 11).

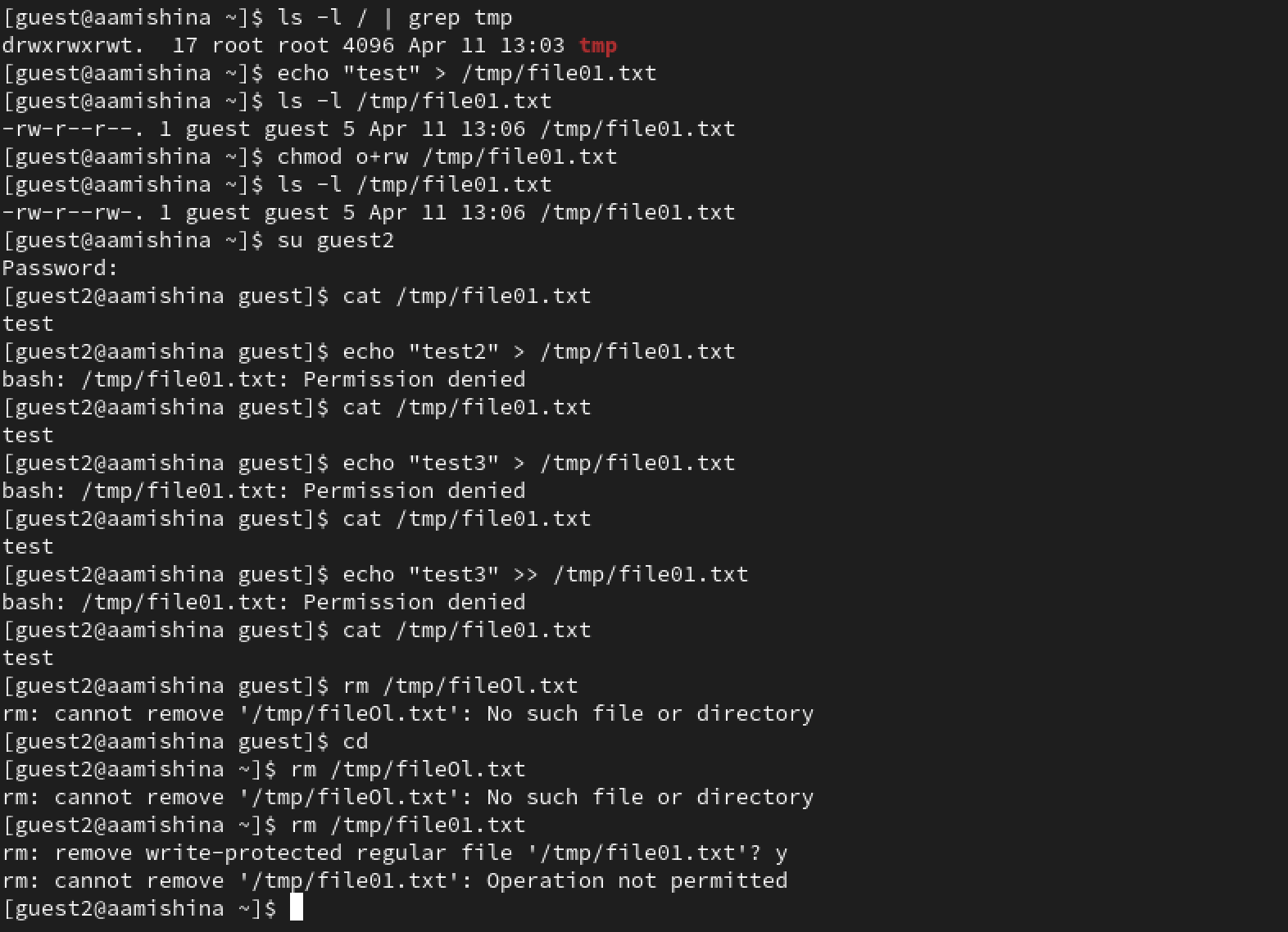


Рис. 11: Проверка наличия атрибута Sticky на /tmp, работа с файлом file01.txt

Повышаем свои права до суперпользователя следующей командой su - и выполняем после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp. Выходим из режима суперпользователя: exit. От пользователя guest2 проверяем, что атрибута t у директории /tmp нет: ls -l / | grep tmp. Повторяем предыдущие шаги. Чтение и удаления удается выполнить, на запись и перезапись получаем отказ (рис. 12).

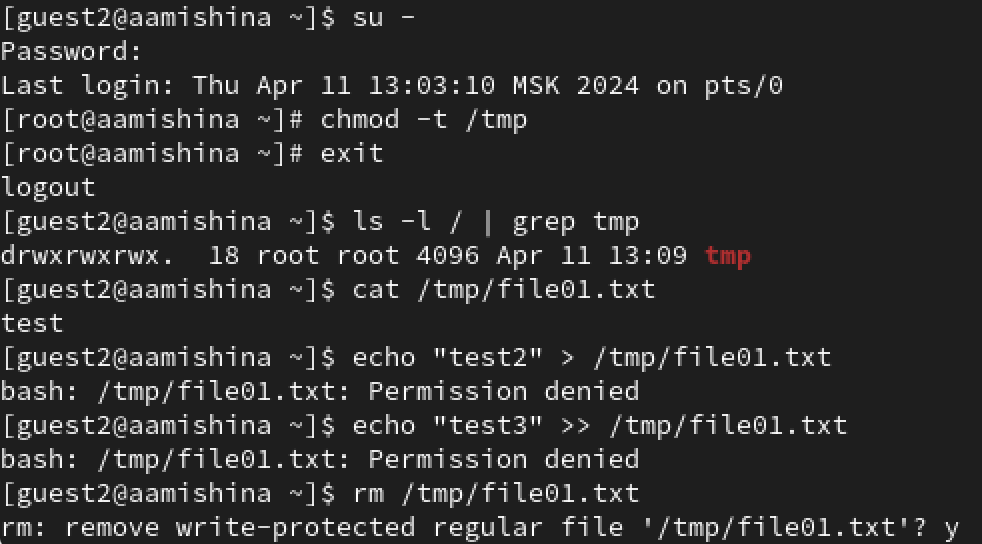


Рис. 12: Снятие атрибута t (Sticky-бит), повторение операций

Повышаем свои права до суперпользователя и возвращаем атрибут t на директорию /tmp: su -, chmod +t /tmp и exit (рис. 13).

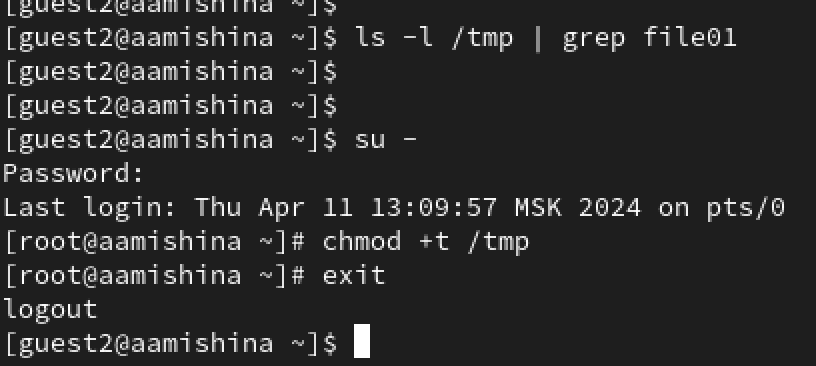


Рис. 13: Возвращение атрибута

# 3 Выводы

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

# Список литературы