Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Аскеров Александр Эдуардович

Содержание

[1 Цель работы 1](#__RefHeading___Toc800_2507336300)

[2 Выполнение лабораторной работы 1](#__RefHeading___Toc802_2507336300)

[2.1 Создание программы 1](#__RefHeading___Toc804_2507336300)

[2.2 Исследование Sticky-бита 8](#__RefHeading___Toc806_2507336300)

[3 Выводы 12](#__RefHeading___Toc808_2507336300)

[Список литературы 12](#__RefHeading___Toc810_2507336300)

# 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.

# 2 Выполнение лабораторной работы

## 2.1 Создание программы

Отключим систему запретов до очередной перезагрузки системы.

Рис. 1: Отключение системы запретов до перезагрузки

Рис. 1: Отключение системы запретов до перезагрузки

Войдём в систему от имени пользователя guest.

Рис. 2: Смена пользователя на guest

Рис. 2: Смена пользователя на guest

Создадим программу simpleid.c.

Рис. 3: Создание программы simpleid.c

Рис. 3: Создание программы simpleid.c

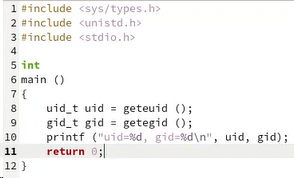


Рис. 4: Программа simpleid.c

Скомпилируем программу и убедимся, что файл программы создан.

Рис. 5: Компиляция программы

Рис. 5: Компиляция программы

Выполним программу simpleid.

Рис. 6: Отработка программы

Рис. 6: Отработка программы

Выполним системную программу id.

Рис. 7: Команда id

Рис. 7: Команда id

Видим, что выводы программы и команды id совпадают.

Усложним программу, добавив вывод действительных идентификаторов.

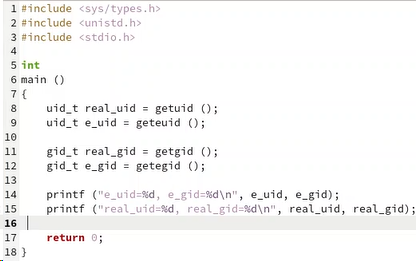


Рис. 8: Программа simpleid2.c

Скомпилируем и запустим simpleid2.c.

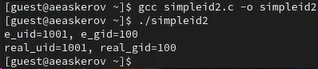


Рис. 9: Компиляция и отработка программы

От имени суперпользователя выполним следующие команды.

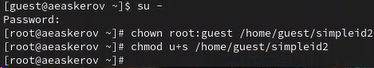


Рис. 10: Выполнение команд chown и chmod

Теперь пользователь root будет владельцем файла, а группа guest будет группой этого файла. Вторая команда устанавливает бит установки SUID для файла simpleid2, позволяющий запускать файл с привилегиями пользователя-владельца файла, а не пользователя, который запускает его.

Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2.

Рис. 11: Проверка

Рис. 11: Проверка

Запустим simpleid2 и id.

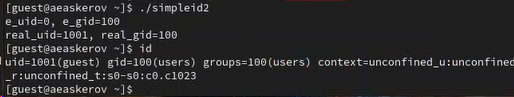


Рис. 12: Запуск simpleid2 и id

Результаты отличаются.

Проделаем тоже самое относительно SetGID-бита.

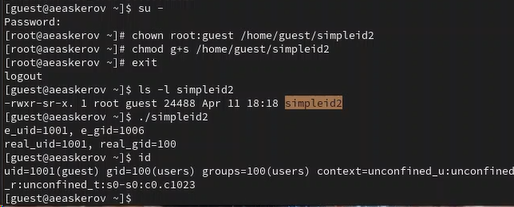


Рис. 13: Повтор действий относительно SetGID-бита

Создадим программу readfile.c.

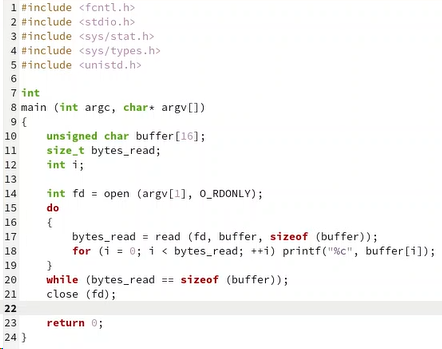


Рис. 14: Программа readfile.c

Откомпилируем её.

Рис. 15: Компиляция программы

Рис. 15: Компиляция программы

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

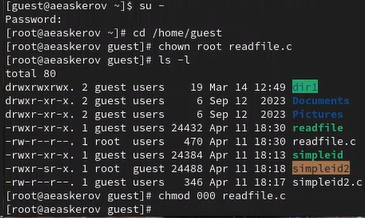


Рис. 16: Смена владельца и изменение прав

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

Рис. 17: Проверка

Рис. 17: Проверка

Сменим у программы readfile владельца и установим SetU’D-бит.

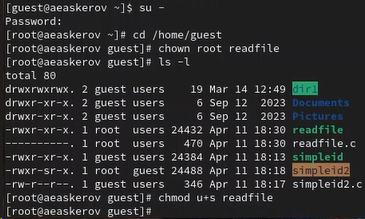


Рис. 18: Смена владельца readfile и установка SetU’D-бита

Проверим, может ли программа readfile прочитать файл readfile.c.

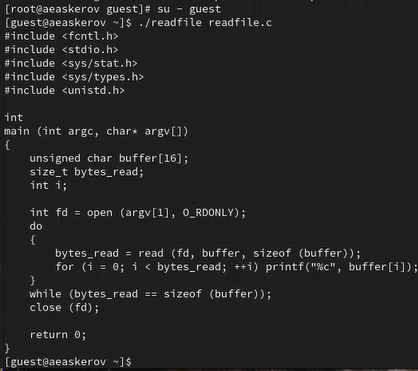


Рис. 19: Проверка

Проверим, может ли программа readfile прочитать файл /etc/shadow.

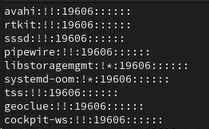


Рис. 20: Проверка

## 2.2 Исследование Sticky-бита

Выясним, установлен ли атрибут Sticky на директории /tmp.

Рис. 21: Проверка

Рис. 21: Проверка

От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test.

Рис. 22: Создание файла file01.txt

Рис. 22: Создание файла file01.txt

Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные».

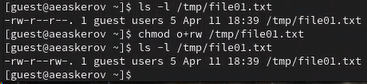


Рис. 23: Изменение прав на file01.txt

От пользователя guest2 (не являющегося владельцем) попробуем прочитать файл /tmp/file01.txt.

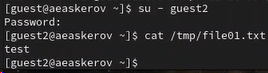


Рис. 24: Чтение файла file01.txt

От пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2.

Рис. 25: Изменение файла file01.txt

Рис. 25: Изменение файла file01.txt

Проверим содержимое файла.

Рис. 26: Проверка

Рис. 26: Проверка

От пользователя guest2 попробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию.

Рис. 27: Попытка дозаписи в файл file01.txt

Рис. 27: Попытка дозаписи в файл file01.txt

Доступ запрещён.

Проверим содержимое файла.

Рис. 28: Проверка

Рис. 28: Проверка

От пользователя guest2 попробуем удалить файл /tmp/file01.txt.

Рис. 29: Попытка удалить файл file01.txt

Рис. 29: Попытка удалить файл file01.txt

Удалить файл не удалось.

Повысим свои права до суперпользователя и выполним после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp.

Рис. 30: Снятие атрибута t с директории /tmp

Рис. 30: Снятие атрибута t с директории /tmp

Покинем режим суперпользователя.

Рис. 31: Смена пользователя

Рис. 31: Смена пользователя

От пользователя guest2 проверим, что атрибута t у директории /tmp нет.

Рис. 32: Проверка

Рис. 32: Проверка

Повторим предыдущие шаги.

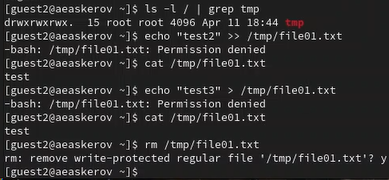


Рис. 33: Повтор действий

Ничего не изменилось, за исключением того, что файл получилось удалить от имени пользователя, не являющегося его владельцем.

Повысим свои права до суперпользователя и вернём атрибут t на директорию /tmp.

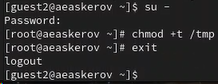


Рис. 34: Возврат атрибута t директории /tmp

# 3 Выводы

Изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами. Рассмотрена работа механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

# Список литературы

1. [How does the sticky bit work](https://unix.stackexchange.com/questions/79395/how-does-the-sticky-bit-work)