Отчёт по лабораторной работе

Лабораторная №5

Панкратьев Александр Владимирович

Содержание

# Цель работы

Изучить механизмы изменения идентификаторов, применения SetUID- и Sticky-биты. Получить практические навыки работы в консоли с дополнительными атрибутами. Рассмотреть работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

# Выполнение лабораторной работы

Для выполнения работы, установил компилятор gcc и отключил защиту SELinux (рис. 1, 2).

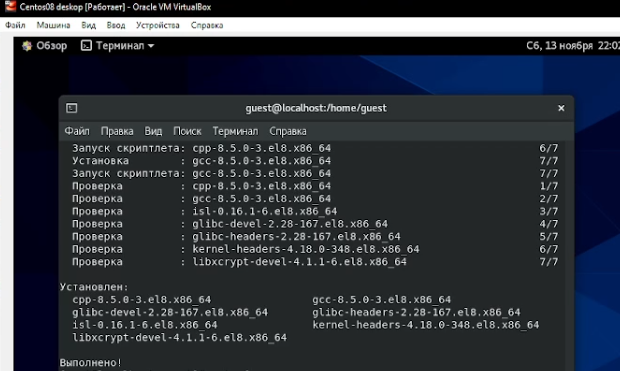


Figure 1: Установка gcc



Figure 2: Снятие ограничений SELinux

## Исследование SetUID- и SetGID-битов

Вошёл в систему от пользователя guest и создал программу simpleid.c (рис. 3).

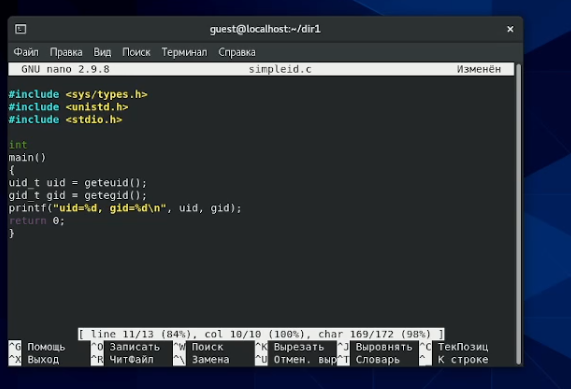


Figure 3: Код программы simpleid.c

Скомпилировал и выполнил программу. Полученный результат совпал с выводом команды id (рис. 4)



Figure 4: Компиляция и выполнение программы simpleid.c

Добавил в программу вывод действительных идентификаторов, назвал ее simpleid2.c (рис. 5).

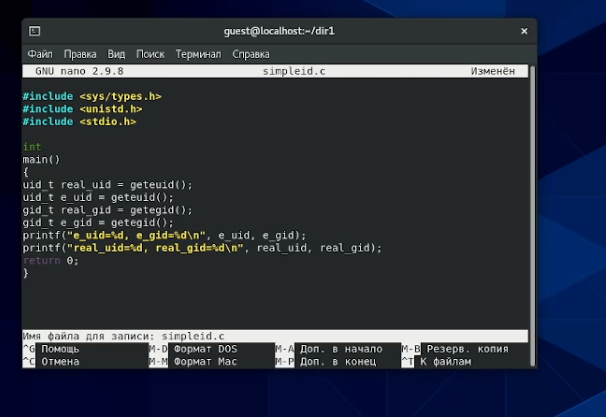


Figure 5: Код программы simpleid2.c

Скомпилировал и запустил программу simpleid2.c. Действительные идентификаторы совпали с эффективными (рис. 6)

Figure 6: Компиляция и выполнение программы simpleid2.c

Figure 6: Компиляция и выполнение программы simpleid2.c

От имени суперпользователя изменила владельца программы simpleid2 на root и добавил атрибут SetUID. (рис. 7)



Figure 7: Изменение атрибутов программы simpleid2

Проверил правильность установки новых атрибутов и смены владельца файла simpleid2 и запустил simpleid2. Теперь вывод программы отличается от вывода команды id. Действительные идентификаторы остались прежними, а эффективный идентификатор пользователя теперь равен 0 - это идентификатор суперпользователя. Это значит, что пользователь guest использует права суперпользователя во время выполнения программы (рис. 8)

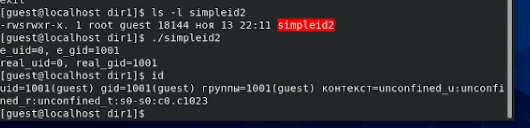


Figure 8: Вывод программы simpleid2 с атрибутом SetUID

Создал программу readfile.c (рис. 9)

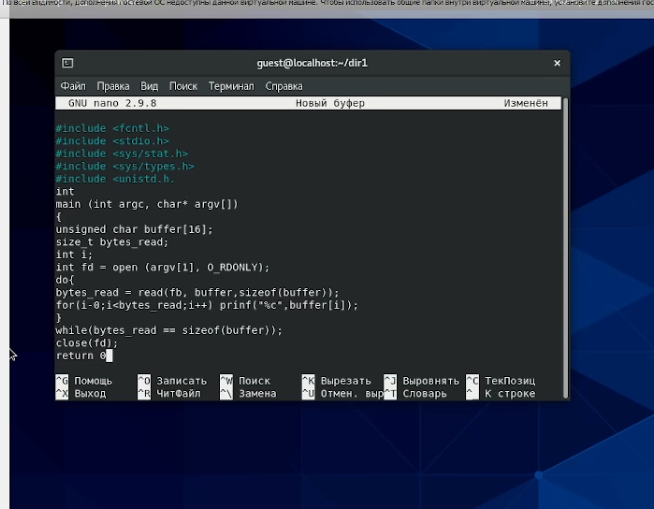


Figure 9: Код программы readfile.c

Откомпилировал и проверил корректность выполения программы (рис. 10)

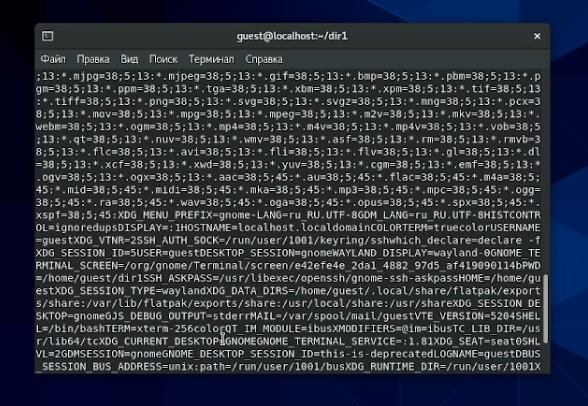


Figure 10: Выполнение программы readfile

Из-за неправильности системы, код не выполнился

Сменил владельца у файла readfile.c и изменил права так, чтобы только суперпользователь мог прочитать его, а guest не мог (рис. 11)

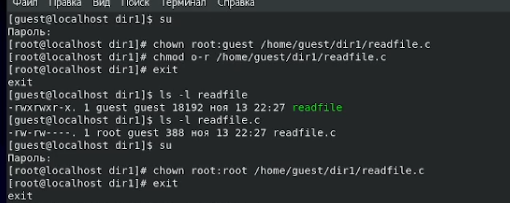


Figure 11: Смена атрибутов файла readfile.c

Проверил, что пользователь guest не может прочитать файл readfile.c (рис. 12)

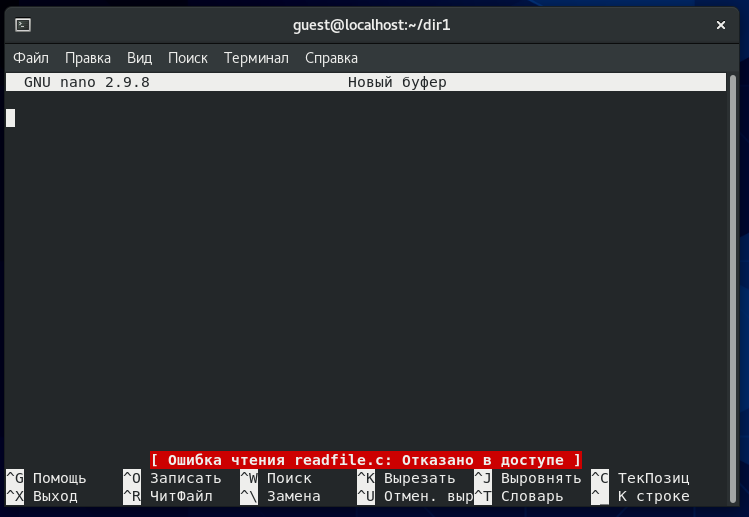


Figure 12: Проверка атрибутов файла readfile.c

Сменил у программы readfile вдадельца на root и установил SetUID-бит (рис. 13)



Figure 13: Добавление SetUID-бита к программе readfile

Теперь с помощью программы readfile можно от имени пользователя guest прочитать файл readfile.c. (рис. 14)

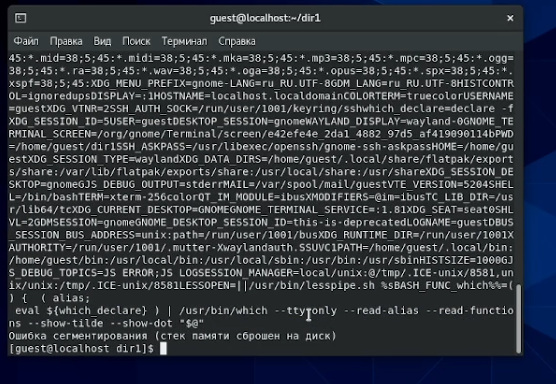


Figure 14: Чтение файла readfile.c с помощью readfile

## Исследование Sticky-бита

Посмотрел, что на директории /tmp установлен атрибут Sticky. От имени пользователя guest создал файл file01.txt в директории /tmp со словом “test”. Посмотрел атрибуты у file01.txt и разрешил чтение и запись для категории пользователей “other” (рис. 17)

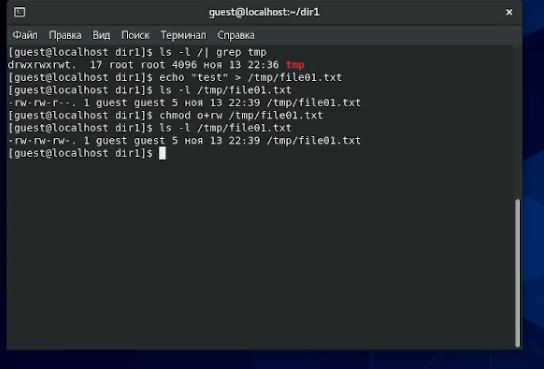


Figure 15: Проверка атрибута Sticky и создание файла в /tmp

От пользователя guest попробовал выполнить различные действия - прочитать файл, дозаписать текст в файл, переписать текст в файле, удалить файл. Получилось сделать все (рис. 16)

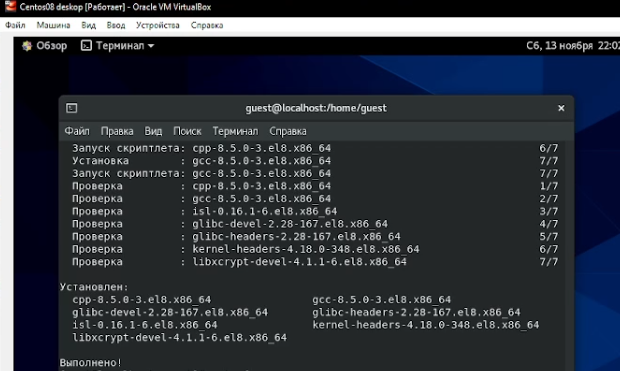


Figure 16: Выполнение операций над file01.txt от имени guest2

От имени суперпользователя снял Sticky-бит с директории /tmp (рис. 17)

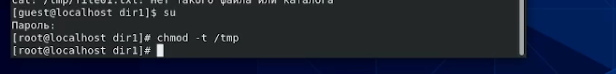


Figure 17: Снятие атрибута Sticky с /tmp

# Вывод

Я изучил механизмы изменения идентификаторов, применения SetUID- и Sticky-биты. Получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.