Отчет по лабораторной работе №5

Дисциплина: Информационная безопасность

Выполнила Дяченко Злата Константиновна, НФИбд-03-18

Содержание

# 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.

# Задание

Создать программы и исследовать SetUID- и Sticky-биты.

# Выполнение лабораторной работы

## Шаг 1

Вошла в систему от имени пользователя guest и создала программу simpleid.c, код которой представлен на Рисунке 1 (рис. 1)

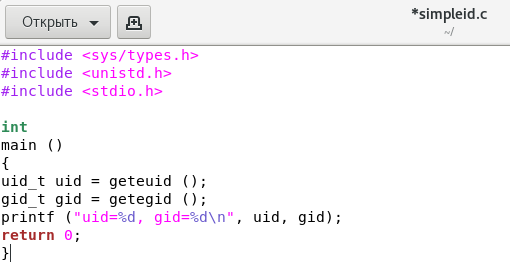


Figure 1: Программа simpleid.c

## Шаг 2

Скомпилировала программу и убедилась, что файл программы создан (рис. 2).

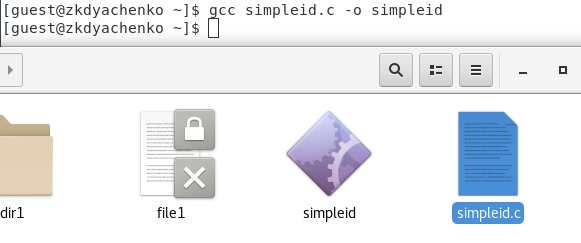


Figure 2: Компиляция программы

## Шаг 3

Выполнила программу simpleid и системную программу id - получила одинаковые результаты (рис. 3).

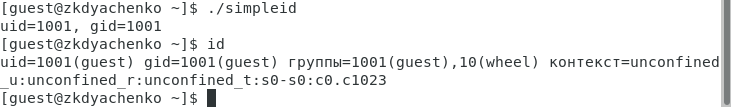


Figure 3: Результат выполнения программы simpleid и системной программы id

## Шаг 4

Усложнила программу, добавив вывод действительных идентификаторов, назвала новую программу simpleid2.c, а ее код предоставлен на Рисунке 4 (рис. 4).

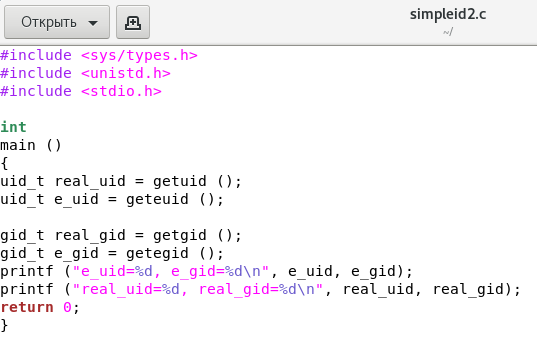


Figure 4: Программа simpleid.c

## Шаг 5

Скомпилировала и запустила simpleid2.c (рис. 5). Теперь мы получаем не только указание на «настоящих» пользователя и группу, «управляющих» процессом, но и на владельца файла.

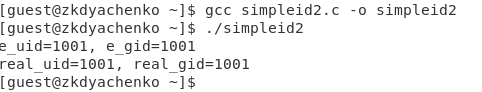


Figure 5: Результат выполнения программы simpleid2

## Шаг 6

От имени суперпользователя выполнила команду chown, чтобы изменить владельца (пользователя и группу) файла simpleid2 и chmod u+s, чтобы установить SetUID (рис. 6).

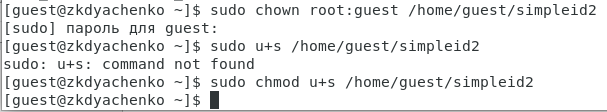


Figure 6: Выполнение команд

## Шаг 7

Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2. А затем запустила simpleid2 и id (рис. 7). Команда id показывает uid и gid текущего пользователя и группы, а программа simpleid2 показывает также uid и gid владельца файла - uid=0 так как файлом владеет пользователь root, группа совпадает с текущей.



Figure 7: Выполнение программы simpleid2 после смены владельца файла

## Шаг 8

Проделала тоже самое относительно SetGID-бита (рис. 8).

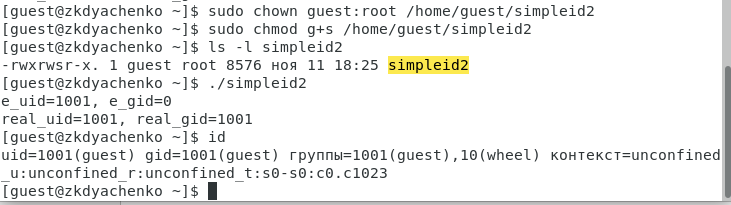


Figure 8: Установка SetGID-бита и выполнение программы simpleid2

## Шаг 9

Создала программу readfile.c (рис. 9).

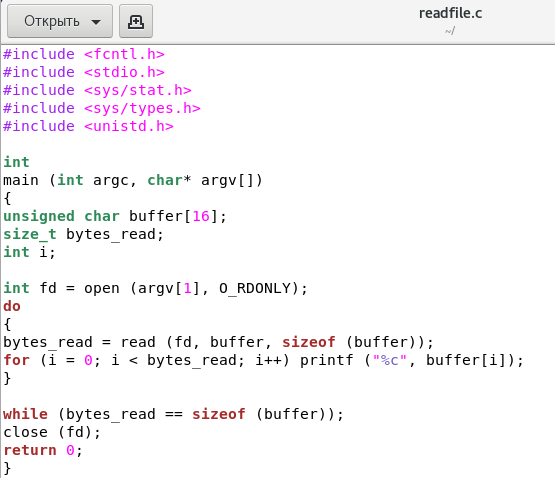


Figure 9: Программа readfile.c

## Шаг 10

Откомпилировала программу. Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог. Проверила, что пользователь guest не может прочитать файл readfile.c (рис. 10).

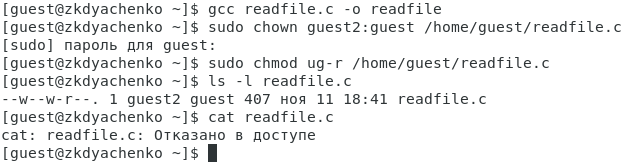


Figure 10: Новый владелец readfile.c

## Шаг 11

Сменила у программы readfile владельца и установила SetUID-бит (рис. 11).

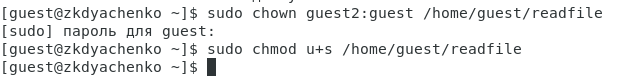


Figure 11: Смена владельца и установка SetUID-бита

## Шаг 12

Проверила, что программа readfile не может прочитать файл readfile.c (рис. 12).



Figure 12: Попытка прочитать readfile.c

## Шаг 13

Проверила, что программа readfile не может прочитать файл /etc/shadow, потому что владелец файла программы - guest2 (рис. 13).

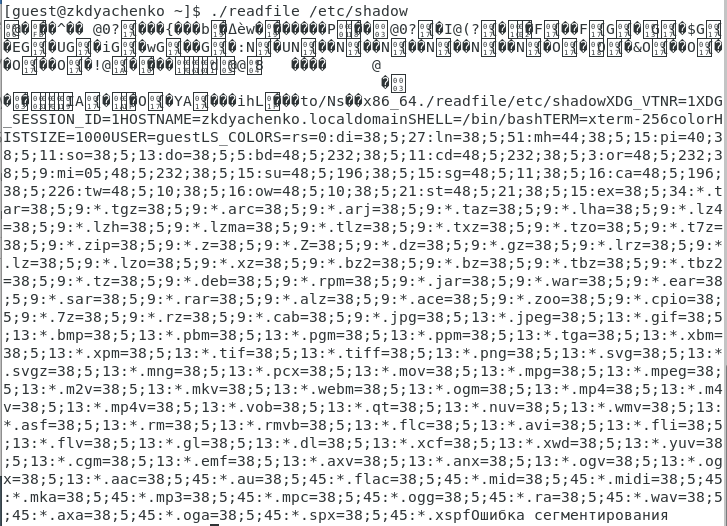


Figure 13: Попытка прочитать /etc/shadow

## Шаг 14

Выяснила, что установлен атрибут Sticky на директории /tmp - в конце списка атрибутов видим t (рис. 14). От имени пользователя guest создала файл file01.txt в директории /tmp со словом test. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные».

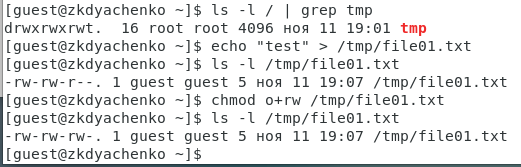


Figure 14: Создание файла и установка атрибутов

## Шаг 15

От пользователя guest2 прочитала файл /tmp/file01.txt. Кроме того, удалось дозаписать в файл слово test2, записать слово test3, стерев при этом всю имеющуюся в файле информацию (рис. 15).

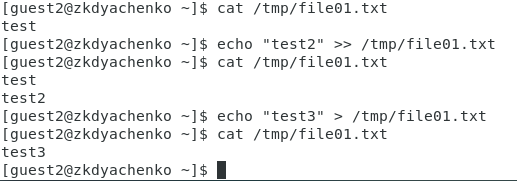


Figure 15: Изменение файла /tmp/file01.txt от имени пользователя guest2

## Шаг 16

От пользователя guest2 попробовала удалить файл /tmp/file01.txt, но сделать это не удалось. Повысила свои права до суперпользователя и выполнила после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp. Затем покинула режим суперпользователя (рис. 16)

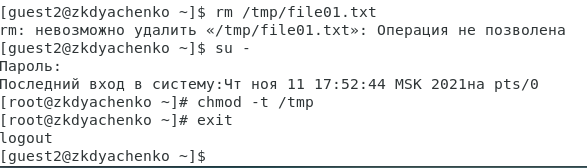


Figure 16: Снятие Sticky-бита

## Шаг 17

От пользователя guest2 проверила, что атрибута t у директории /tmp нет. От пользователя guest2 прочитала файл /tmp/file01.txt, дозаписала в файл слово test2, записала слово test3, стерев при этом всю имеющуюся в файле информацию. Кроме того, удалось удалить файл от имени guest2, хотя он не был владельцем файла (рис. 17).

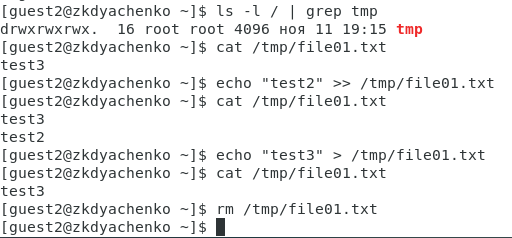


Figure 17: Удаление файла от имени guest2

## Шаг 18

Повысила свои права до суперпользователя и вернула атрибут t на директорию /tmp (рис. 18).

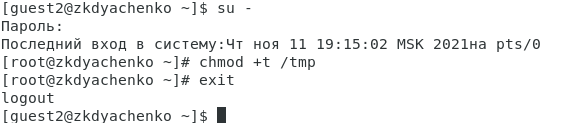


Figure 18: Возвращение атрибута t

# Выводы

В результате работы я изучила механизмы изменения идентификаторов, применение SetUID- и Sticky-битов. Были получены практические навыки работы в консоли с дополнительными атрибутами и рассмотрена работа механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов. Результаты работы находятся в [репозитории на GitHub](https://github.com/ZlataDyachenko/workD), а также есть [скринкаст выполнения лабораторной работы](https://www.youtube.com/watch?v=OipLW7aKabU).