Отчёт по лабораторной работе 5

Соболев Максим Сергеевич

Содержание

# 1 Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

# 2 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.

# 3 Задание

Исследовать влияние дополнительных атрибутов Исследовать Sticky-бит

# 4 Теоретическое введение

В Linux существует три основных вида прав — право на чтение, запись и выполнение, а также три категории пользователей, к которым они могут применяться — владелец файла, группа владельца и все остальные. \*\*\* # Выполнение лабораторной работы часть 1

## 4.1 Шаг 1

Входим в систему от имени пользователя guest.

Рис. 1: 1

Рис. 1: 1

## 4.2 Шаг 2

Создаем программу



Рис. 2: 2

## 4.3 Шаг 3

Комплилируем программу и убеждаемся, что файл программы создан: gcc simpleid.c -o simpleid

Рис. 3: 3

Рис. 3: 3

## 4.4 Шаг 4

Выполняем программу simpleid:./simpleid

4 ## Шаг 5

Выполняем системную программу id: id и сравниваем полученный результат с данными предыдущего пункта задания.

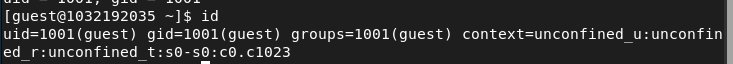


Рис. 4: 5

## 4.5 Шаг 6

Усложняем программу, добавив вывод действительных идентификаторов

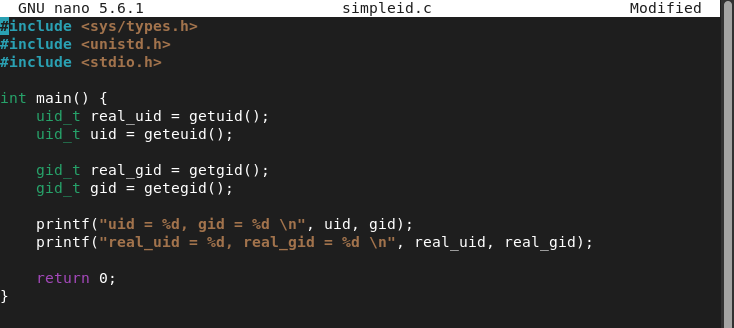


Рис. 5: 6

Получившуюся программу называем simpleid2.c.

## 4.6 Шаг 7

Компилируем и запускаем simpleid2.c: gcc simpleid2.c -o simpleid2 ./simpleid2

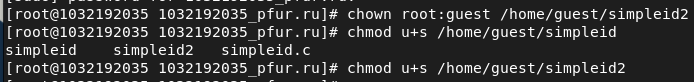


Рис. 6: 7

## 4.7 Шаг 8

От имени суперпользователя выполняем команды: chown root:guest /home/guest/simpleid2 chmod u+s /home/guest/simpleid2

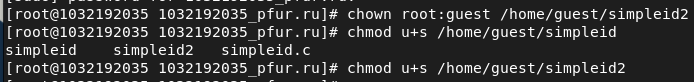


Рис. 7: 8

## 4.8 Шаг 9

Выполняем проверку правильности установки новых атрибутов и смены владельца файла simpleid2: ls -l simpleid2

Рис. 8: 10

Рис. 8: 10

## 4.9 Шаг 10

Запускаем simpleid2 и id:./simpleid2 id и сравниваем результаты

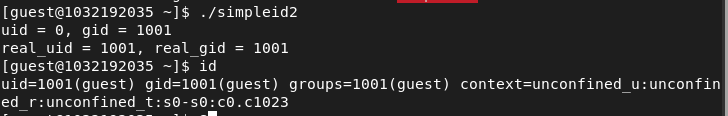


Рис. 9: 11

## 4.10 Шаг 11

Проделываем тоже самое относительно SetGID-бита

Рис. 10: 12.1

Рис. 10: 12.1

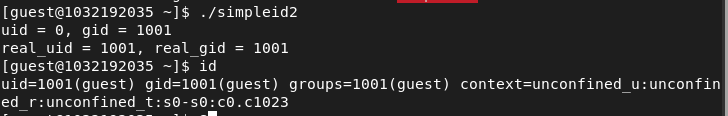


Рис. 11: 12.2

## 4.11 Шаг 12

Создаем программу readfile.c:

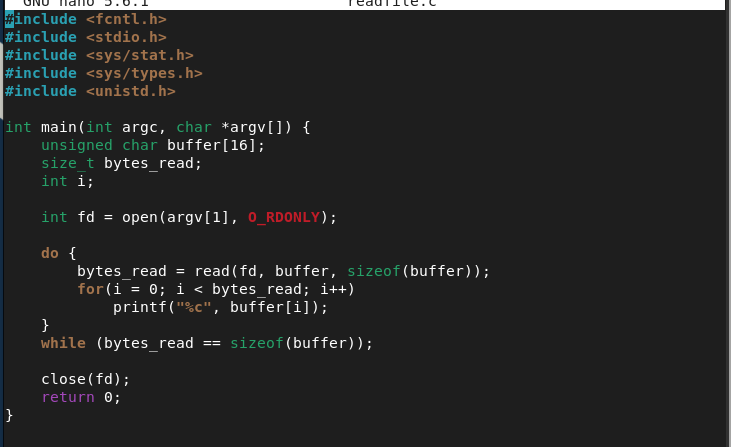


Рис. 12: 13

## 4.12 Шаг 13

Компилируем её. gcc readfile.c -o readfile

Рис. 13: 14

Рис. 13: 14

## 4.13 Шаг 14

Сменяем владельца у файла readfile.c и изменяем права так, чтобы только суперпользователь мог прочитать его, a guest не мог

Рис. 14: 15

Рис. 14: 15

## 4.14 Шаг 15

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

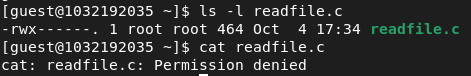


Рис. 15: 16

Получившуюся программу называем simpleid2.c.

## 4.15 Шаг 16

Меняем у программы readfile владельца и устанавливаем SetU’D-бит.

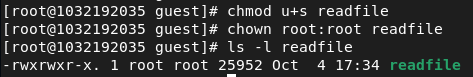


Рис. 16: 17

## 4.16 Шаг 17

Проверяем, может ли программа readfile прочитать файл readfile.c?

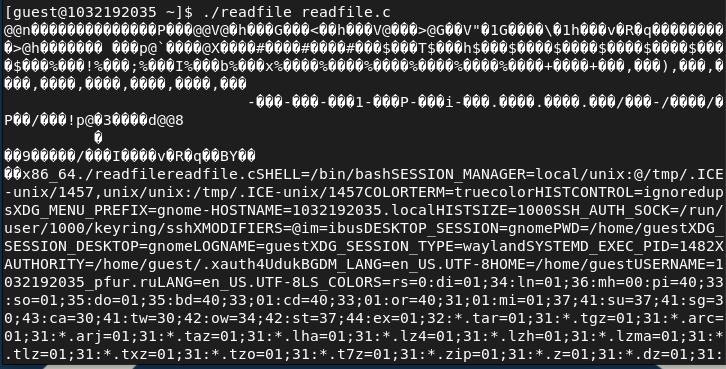


Рис. 17: 18

Чтение файла невозможно, программа выдаёт segfault при попытке чтения файлов

## 4.17 Шаг 18

Проверяем, может ли программа readfile прочитать файл /etc/shadow?

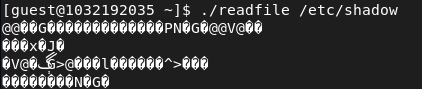


Рис. 18: 19

Чтение файла невозможно, программа выдаёт segfault при попытке чтения файлов. При этом попытка чтения файла без атрибута +s, принадлежащего пользователю guest происходит без проблем \*\*\* # Выполнение лабораторной работы часть 2

## 4.18 Шаг 1

Выясняем, установлен ли атрибут Sticky на директории /tmp, для чего выполняем команду ls -l / | grep tmp

Рис. 19: 2.1

Рис. 19: 2.1

## 4.19 Шаг 2

От имени пользователя guest создаем файл file01.txt в директории /tmp со словом test: echo “test” > /tmp/file01.txt

Рис. 20: 2.2

Рис. 20: 2.2

## 4.20 Шаг 3

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 ls -l /tmp/file01.txt

Рис. 21: 2.3

Рис. 21: 2.3

## 4.21 Шаг 4

От пользователя guest2 попробуем прочитать файл /tmp/file01.txt: cat /tmp/file01.txt

2.4 ## Шаг 5

От пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2 командой echo “test2” > /tmp/file01.txt

Рис. 22: 2.5

Рис. 22: 2.5

## 4.22 Шаг 6

Проверьте содержимое файла командой cat /tmp/file01.txt

Рис. 23: 2.6

Рис. 23: 2.6

## 4.23 Шаг 7

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

Рис. 24: 2.7

Рис. 24: 2.7

## 4.24 Шаг 8

Проверяем содержимое файла командой cat /tmp/file01.txt

Рис. 25: 2.8

Рис. 25: 2.8

## 4.25 Шаг 9

От пользователя guest2 попробуем удалить файл /tmp/file01.txt командой rm /tmp/fileOl.txt

Рис. 26: 2.9

Рис. 26: 2.9

## 4.26 Шаг 10

Повышаем свои права до суперпользователя следующей командой su - и выполняем после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp

Рис. 27: 2.10

Рис. 27: 2.10

## 4.27 Шаг 11

Покидаем режим суперпользователя командой exit

Рис. 28: 2.11

Рис. 28: 2.11

## 4.28 Шаг 12

От пользователя guest2 проверяем, что атрибуты t у директории /tmp нет: ls -l / | grep tmp

Рис. 29: 2.12

Рис. 29: 2.12

## 4.29 Шаг 13

Запись и дозапись работают без изменений, удаление файла стало доступно

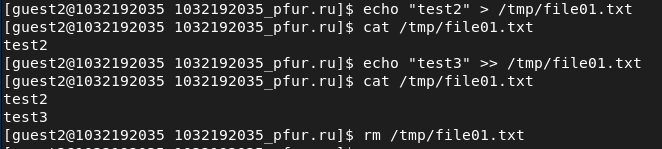


Рис. 30: 2.13

## 4.30 Шаг 14

Да удалось

## 4.31 Шаг 15

Повышаем свои права до суперпользователя и возвращаем атрибут t на директорию /tmp: su - chmod +t /tmp exit

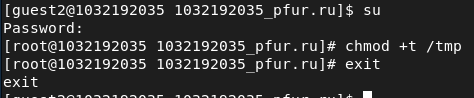


Рис. 31: 2.15

# 5 Выводы

Мы изучили механизмы изменения идентификаторов, применение SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Рассмотрели работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

# Список литературы

1. xattr(7) — Linux manual page // Linux man-pages project URL: https://man7.org/linux/man-pages/man7/xattr.7.html (дата обращения: 30.09.2022).