**Лабораторная работа N° 5**

Информационная безопасность

Леон Фернандо Хосе Фернандо | НПМбд02-20

**Содержание**

[**1 Цель работы** 4](#_Toc147541021)

[**2 Теоретическое введение** 4](#_Toc147541022)

[**3 Выполнение лабораторной работы** 4](#_Toc147541023)

[**3.1 Создание программы** 4](#_Toc147541024)

[**3.2Исследование Sticky-бита** 10](#_Toc147541025)

[**4 Выводы** 12](#_Toc147541026)

[**5 Список Литературы** 12](#_Toc147541027)

**Список иллюстраций**

[Рисунок 1. Предварительная подготовка 5](#_Toc147541074)

[Рисунок 2. Команда “whereis” 5](#_Toc147541075)

[Рисунок 3.Вход в систему и создание программы 6](#_Toc147541076)

[Рисунок 4. Код программы simpleid.c 6](#_Toc147541077)

[Рисунок 5. Компиляция и выполнение программы simpleid 6](#_Toc147541078)

[Рисунок 6. Усложнение программы 7](#_Toc147541079)

[Рисунок 7. Переименование программы в simpleid2.c 7](#_Toc147541080)

[Рисунок 8. Компиляция и выполнение программы simpleid2 7](#_Toc147541081)

[Рисунок 9. Установка новых атрибутов (SetUID) 8](#_Toc147541082)

[Рисунок 10. Запуск simpleid2 после установки SetUID 8](#_Toc147541083)

[Рисунок 11. Запуск simpleid2 после установки SetGID 8](#_Toc147541084)

[Рисунок 12. Код программы readfile.c 9](#_Toc147541085)

[Рисунок 13. Смена владельца и прав доступа у файла readfile.c 9](#_Toc147541086)

[Рисунок 14. Запуск программы readfile 10](#_Toc147541087)

[Рисунок 15. Создание файла file01.txt 10](#_Toc147541088)

[Рисунок 16. Попытка выполнить действия над файлом file01.txt от имени пользо. 11](#_Toc147541089)

[Рисунок 17. Удаление атрибута t (Sticky-бита) 11](#_Toc147541090)

[Рисунок 18. Возвращение атрибута t (Sticky-бита) 12](#_Toc147541091)

# **1 Цель работы**

Изучение механизмов модификации идентификатора, использования SetUID и Sticky bits. Получение практических навыков использования консольных команд с дополнительными атрибутами. Изучается работа механизма изменения идентификатора пользовательского процесса, а также влияние бита Sticky на запись и удаление файлов.

# **2 Теоретическое введение**

SetUID, SetGID и Sticky - это специальные типы разрешений позволяют задавать

расширенные права доступа на файлы или каталоги.

• SetUID (set user ID upon execution — «установка ID пользователя во время

выполнения) являются флагами прав доступа в Unix, которые разрешают

пользователям запускать исполняемые файлы с правами владельца испол-

няемого файла.

• SetGID (set group ID upon execution — «установка ID группы во время вы-

полнения») являются флагами прав доступа в Unix, которые разрешают

пользователям запускать исполняемые файлы с правами группы исполняе-

мого файла.

• Sticky bit в основном используется в общих каталогах, таких как /var или

/tmp, поскольку пользователи могут создавать файлы, читать и выполнять

их, принадлежащие другим пользователям, но не могут удалять файлы,

принадлежащие другим пользователям.

# **3 Выполнение лабораторной работы**

## **3.1 Создание программы**

Для начала я проверил наличие компилятора gcc, используя команду 'gcc -v'. Затем я отключил системные ограничения до следующей перезагрузки системы с помощью команды "sudo setenforce 0", после чего команда "getenforce" отобразила "Разрешающий".

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок . Предварительная подготовка

Проверить успешное выполнение команд “whereis gcc” и “whereis g++” (их расположение).

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок . Команда “whereis”

Я вошел в систему как пользователь "гость", используя команду "su - guest". Я создал программу под названием "simplied.c" с помощью команды "touch simple id.c" и открыл ее в редакторе, используя команду "gedit /home/guest/simpleid.c".

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок .Вход в систему и создание программы

Код программы выглядит следующим образом.

Изображение выглядит как текст, программное обеспечение, снимок экрана

Автоматически созданное описание

Рисунок . Код программы simpleid.c

Я скомпилировал программу и убедился, что программный файл был создан с помощью команды 'gcc simplied.co simpleid". Я выполнил программу 'simpleid' с помощью команды './simpleid', а затем запустил системную программу 'id' с помощью команды 'id'. Результаты, полученные в результате выполнения обеих команд, совпали (uid=1001 и gid=1001).

Изображение выглядит как текст, программное обеспечение, Шрифт, веб-страница

Автоматически созданное описание

Рисунок . Компиляция и выполнение программы simpleid

Усложнил программу, добавив вывод действительных идентификаторов.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок . Усложнение программы

Получившуюся программу назвала simpleid2.c

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок . Переименование программы в simpleid2.c

Скомпилировала и запустила simpleid2.c командами “gcc simpleid2.c -o sipleid2”

и “./simpleid2”.

Изображение выглядит как текст, Шрифт, число, снимок экрана

Автоматически созданное описание

Рисунок . Компиляция и выполнение программы simpleid2

От имени суперпользователя я выполнил команды "sudo chown root:guest /home/guest/simpleid2" и "sudo chmod u+s /home/guest/simpleid2". После этого я проверил правильность установки новых атрибутов и смену владельца файла 'simpleid2', используя команду 'sudo ls -l /home/guest/simpleid2'. Эти команды привели к изменению пользователя файла на "root" и установке бита SetUID.

Изображение выглядит как текст, программное обеспечение, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок . Установка новых атрибутов (SetUID)

Запустил программы simpleid2 и id. Теперь появились различия в uid

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок . Запуск simpleid2 после установки SetUID

Проделал тоже самое относительно SetGID-бита. Также можем заметить раз-

личия с предыдущим пунктом.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок . Запуск simpleid2 после установки SetGID

Создаем программу readfile.c

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок . Код программы readfile.c

Я скомпилировал созданную программу, используя команду 'gcc readfile.c -o readfile'. Я изменил владельца файла 'readfile.c' с помощью команды 'sudo chown root:guest /home /гость / readfile.c' и настроил разрешения так, чтобы только суперпользователь мог его прочитать, в то время как 'гость' не мог, используя команду 'sudo chmod 700 /home/гость/readfile.c'. Теперь я убедился, что пользователь "гость" не может прочитать файл "read file.c" с помощью команды "cat readfile.c" и получил сообщение об ошибке "отказано в доступе".

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок . Смена владельца и прав доступа у файла readfile.c

Я сменил владельца программы "readfile" и установил SetUID. Я проверил, может ли программа "readfile" прочитать файл "read file.c", используя команду "./readfile readfile.c". Он смог это прочитать. Аналогично, я проверил, возможно ли прочитать файл '/etc/shadow', и это также прошло успешно.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок . Запуск программы readfile

## **3.2Исследование Sticky-бита**

Я подтвердил, что атрибут Sticky был установлен в каталоге '/tmp', используя команду 'ls -l / | grep tmp'. От имени пользователя "гость" я создал файл с именем 'file01.txt ' в каталоге '/tmp' со словом 'test' с помощью команды 'echo "test" > /tmp/file01.txt '. Я проверил атрибуты вновь созданного файла и предоставил разрешения на чтение и запись для категории пользователей "все остальные", используя команды "ls -l /tmp/file01.txt ' и 'chmod o+rw /tmp/file01.txt '.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок . Создание файла file01.txt

От имени пользователя "guest2" я попытался прочитать файл, используя команду "cat /tmp/file01.txt ", и это было успешно. Затем я попытался добавить слово "test2" к файлу, проверить его содержимое и записать в файл "test3", удалив при этом всю существующую информацию. Эти операции были успешными только тогда, когда я дополнительно предоставил разрешения на чтение и запись для "группы" пользователей, используя команду "chmod g+rw /tmp/file01.txt '. Однако, когда я попытался удалить файл от имени пользователя 'guest2', это не было возможно ни в одном из случаев, и произошла ошибка.

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Автоматически созданное описание

Рисунок . Попытка выполнить действия над файлом file01.txt от имени пользо.

Я повысил разрешения до статуса суперпользователя с помощью команды 'su -' и выполнил команду для удаления атрибута 't' из каталога '/tmp' с помощью 'chmod -t /tmp'. После этого я вышел из режима суперпользователя с помощью команды "exit". Я повторил предыдущие шаги. Теперь я смог удалить 'file01.txt " файл от имени пользователя, который не был его владельцем.

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок . Удаление атрибута t (Sticky-бита)

Повысила свои права до суперпользователя и вернула атрибут t на директорию

/tmp

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок . Возвращение атрибута t (Sticky-бита)

# **4 Выводы**

Во время выполнения этого лабораторного задания я изучил механизмы модификации идентификаторов, применение SetUID и Sticky bits, а также приобрел практические навыки использования консоли с дополнительными атрибутами. Я изучил работу механизма изменения идентификатора пользовательского процесса и влияние Sticky-бита на запись и удаление файлов.

# **5 Список Литературы**

1. Стандартные права SetUID, SetGID, Sticky в Linux [Электронный ресурс].

URL: https://linux-notes.org/standartny-e-prava-unix-suid-sgid-sticky-bity/.