Отчет по лабораторной работе №5

Информационная безопасность

Чекалова Лилия Руслановна

Содержание

# 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 Задание

* Написание программ
* Изменение владельца файлов и прав доступа на файлы
* Установка и снятие Sticky-бита и проверка доступных действий

# 3 Теоретическое введение

Setuid –– это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо. Классический пример использования этого бита в операционной системе это команда sudo.

На месте, где обычно установлен классический бит x (на исполнение), выставлен специальный бит s. Это позволяет обычному пользователю системы выполнять команды с повышенными привилегиями без необходимости входа в систему как root, зная пароль пользователя root.

Принцип работы Setgid очень похож на setuid с отличием, что файл будет запускаться пользователем от имени группы, которая владеет файлом.

Последний специальный бит разрешения – это Sticky Bit . В случае, если этот бит установлен для папки, то файлы в этой папке могут быть удалены только их владельцем. Пример использования этого бита в операционной системе это системная папка /tmp . Эта папка разрешена на запись любому пользователю, но удалять файлы в ней могут только пользователи, являющиеся владельцами этих файлов.

Более подробно о см. в [1,2].

# 4 Выполнение лабораторной работы

В качестве первого шага лабораторной работы мы осуществили вход от лица пользователя guest и создали файл simpleid.c (рис. [1](#fig:001)).

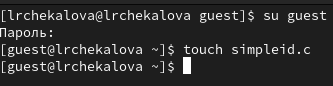


Figure 1: Создание файла

Далее мы написали программу, которая выводитинформацию об идентификаторах пользователя и группы (рис. [2](#fig:002)). Скомпилировали эту программу (рис. [3](#fig:003)).

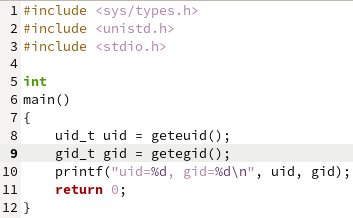


Figure 2: Программа 1

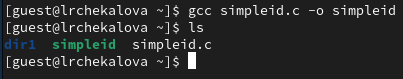


Figure 3: Компиляция программы 1

Выполнили эту программу и сравнили ее вывод с результатом работы системной команды id (рис. [4](#fig:004)). Результаты сходятся, но команда id дает дополнительные сведения, например, название группы.

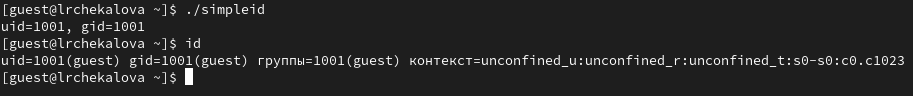


Figure 4: Запуск программы 1

Модифицировали программу, чтобы она выводила также действительные идентификаторы (рис. [5](#fig:005)).

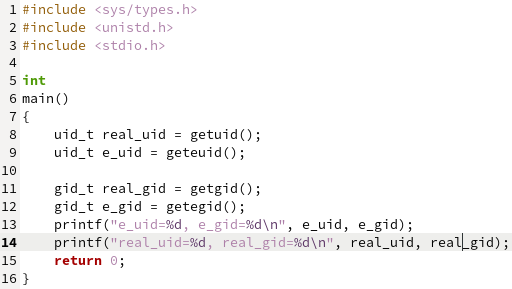


Figure 5: Программа 2

Скомпилировали и запустили модифицированную программу (рис. [6](#fig:006)).

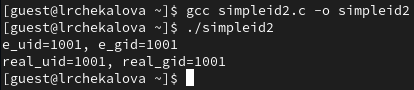


Figure 6: Запуск модифицированной программы

Передали право владения программой суперпользователю и наделили его правом на исполнение программы. Запустили программу и сверили результат с выводом команды id (рис. [7](#fig:007)). Теперь в качестве идентификаторов указана информация root-пользователя.

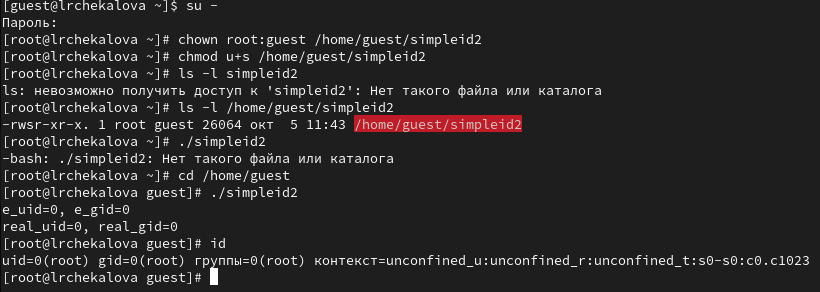


Figure 7: Передача прав и перезапуск программы

Проделали то же самое относительно SetGID-бита (рис. [8](#fig:008)).

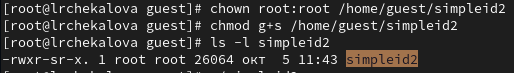


Figure 8: Повтор действий относительно SetGID-бита

Создали новую программу readfile, позволяющую прочесть содержимое файла (рис. [9](#fig:009)).

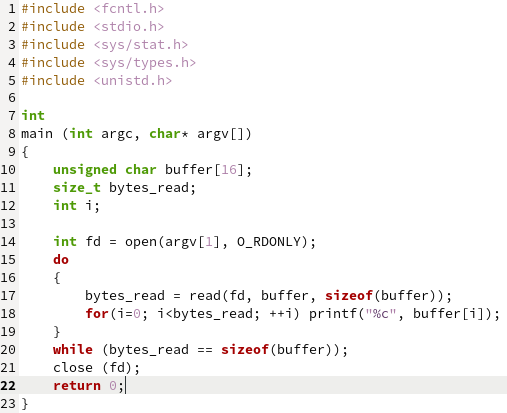


Figure 9: Программа 3

Скомпилировали ее, а затем поменяли владельца файла readfile.c (рис. [10](#fig:010)) и изменили права так, чтобы читать файл мог только суперпользователь. Проверили, может ли пользователь guest прочитать файл (рис. [11](#fig:011)). Ему было отказано в доступе.

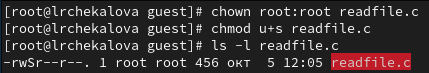


Figure 10: Смена владельца файла

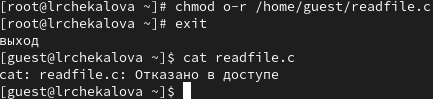


Figure 11: Смена прав и проверка

Передали право владения программой суперпользователю и проверили, может ли программа прочитать readfile.c (рис. [12](#fig:012)) и etc/shadow (рис. [13](#fig:013)). Действия были выполнены успешно.

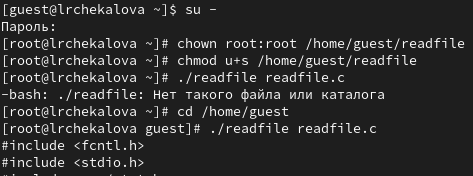


Figure 12: Смена владельца программой и чтение readfile.c

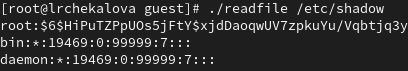


Figure 13: Чтение etc/shadow

Проверили, установлен ли Sticky-бит на директории tmp, создали file01.txt и передали категории пользователей “все остальные” право на чтение и запись (рис. [14](#fig:014)).



Figure 14: Проверка наличия Sticky-бита и создание файла

От лица пользователя guest2 (не являющегося владельцем файла) попробовали прочесть файл и записать туда новые данные (рис. [15](#fig:015)), а также удалить файл (рис. [16](#fig:016)). Мы смогли прочесть файл, но не смогли изменить его и удалить, так как guest2 входит в группу guest, а не в группу “все остальные”.

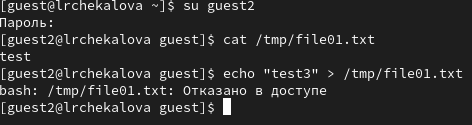


Figure 15: Попытка чтения и изменения

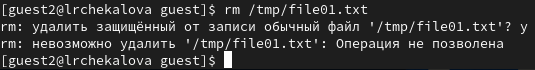


Figure 16: Попытка удаления

Сняли с папки tmp атрибут Sticky (рис. [17](#fig:017)).

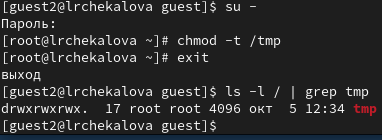


Figure 17: Снятие атрибута Sticky

От лица пользователя guest2 попробовали прочесть файл, произвести в него запись и удалить (рис. [18](#fig:018)). Удаление файла прошло успешно, так как с папки был снят атрибут, запрещавший удаление файлов всеми, кроме владельца.

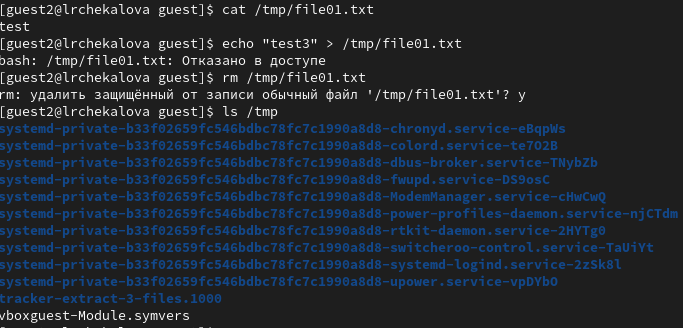


Figure 18: Чтение, запись и удаление

Установили атрибут Sticky обратно на папку tmp (рис. [19](#fig:019)).

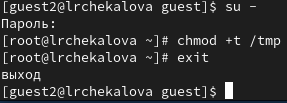


Figure 19: Установка атрибута Sticky

# 5 Выводы

В результате лабораторной работы я повысила навыки использования командой строки и изучила SetUID-, SetGID- и Sticky-биты. Также я рассмотрела работу механизма смены идентификатора процессов пользователей.

# Список литературы

1. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов [Электронный ресурс]. URL: <https://esystem.rudn.ru/mod/resource/view.php?id=1031377>.

2. Использование SETUID, SETGID и Sticky Bit для расширенной настройки прав доступа в операционных системах Linux [Электронный ресурс]. URL: <https://ruvds.com/ru/helpcenter/suid-sgid-sticky-bit-linux/>.