Лабораторная работа № 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Абдуллаев Сайидазизхон Шухратович

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

Закрепить дискреционное разграничение прав в Linux с дополнительными атрибутами.

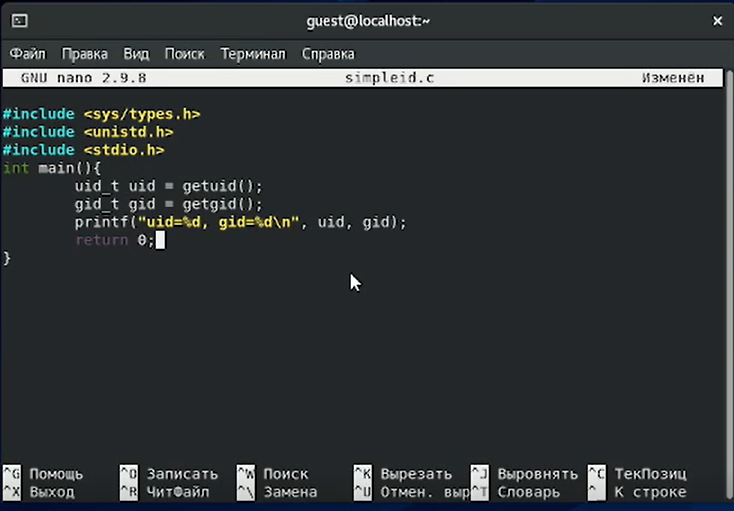
# Теоретическое введение

В Linux, как и в любой многопользовательской системе, абсолютно естественным образом возникает задача разграничения доступа субъектов — пользователей к объектам — файлам дерева каталогов. Один из подходов к разграничению доступа — так называемый дискреционный - предполагает назначение владельцев объектов, которые по собственному усмотрению определяют права доступа субъектов (других пользователей) к объектам (файлам), которыми владеют. Дискреционные механизмы разграничения доступа используются для разграничения прав доступа процессов как обычных пользователей, так и для ограничения прав системных программ в (например, служб операционной системы), которые работают от лица псевдопользовательских учетных записей. Чтобы получить доступ к файлам в Linux, используются разрешения. Эти разрешения назначаются трем объектам: файлу, группе и другому объекту. Для управления правами используется команда chmod. При использовании chmod в относительном режиме вы работаете с тремя индикаторами, чтобы указать, что вы хотите сделать. Сначала вы указываете, для кого вы хотите изменить разрешения. Для этого вы можете выбрать между пользователем (u), группой (g) и другими (o). Затем вы

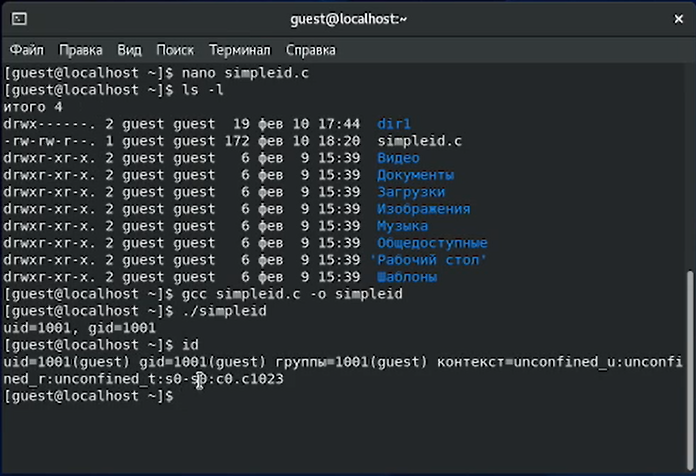
используете оператор для добавления или удаления разрешений из текущего режима или устанавливаете их абсолютно. В конце вы используете r(read), w(write) и x(execute), чтобы указать, какие разрешения вы хотите установить.При использовании chmod вы можете устанавливать разрешения для пользователя (user), группы (group) и других (other).Помимо основных разрешений, о которых вы только что прочитали, в Linux также есть набор расширенных разрешений. Это не те разрешения, которые вы устанавливаете по умолчанию, но иногда они предоставляют полезное дополнение.

# Ход работы

1. Готовим систему и входим из-под пользователя guest. Пишем программу simpleid.c. Компилируем программу, запускаем, видим вывод uid и gid пользователя, сравниваем вывод с id (все совпадает). (Рис. 1, 2).

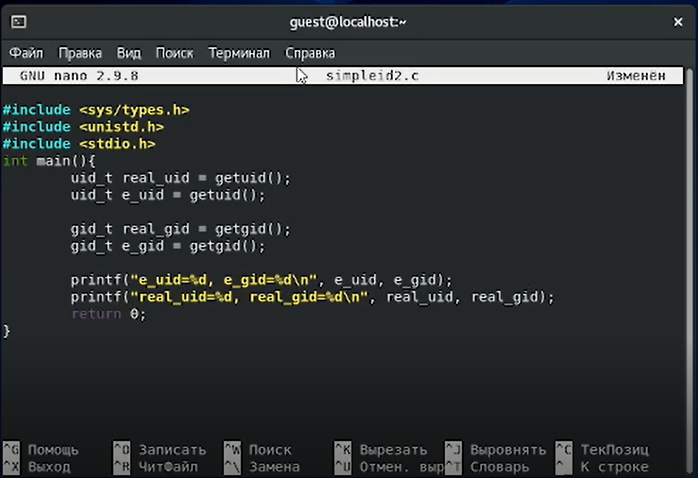


Листинг simpleid.c

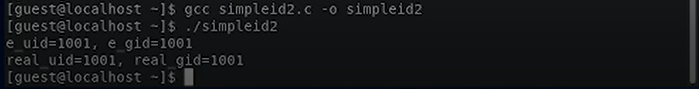


Запуск и сверка simpleid.c

1. Усложняем программу и запускаем её. (Рис. 3, 4).

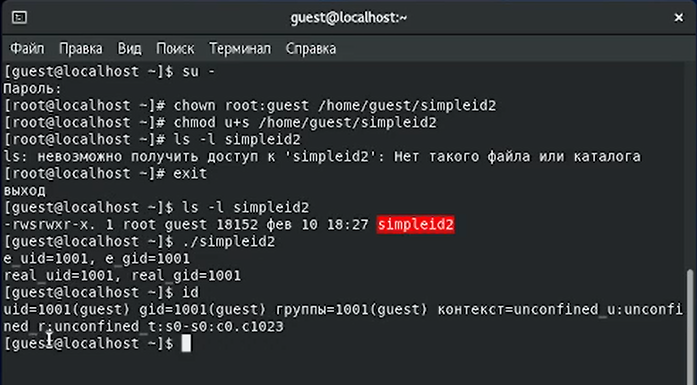


Листинг simpleid2.c



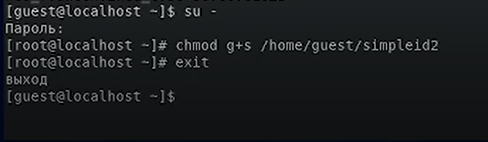
Запуск simpleid2.c

1. Из-под суперпользователя меняем владельца и добавляем SetUID бит на файл. Проверяем правильность и запускаем программу еще раз. еuid возвращает id владельца, а real\_uid возвращает uid запускающего пользователя.(Рис. 5).

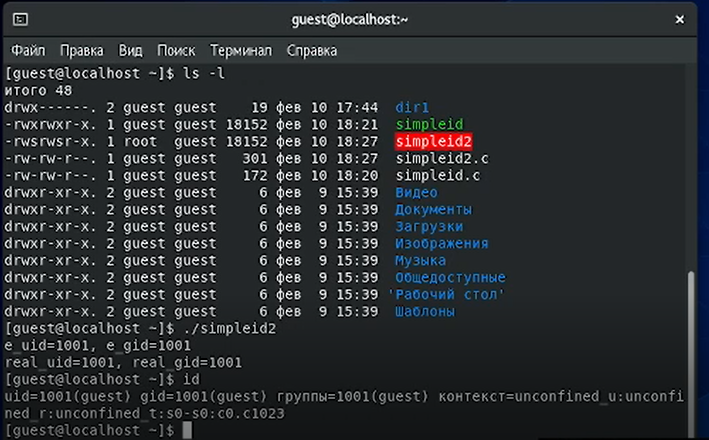


Смена владельца и добавление SetUID бита, запуск и сверка simpleid2.c

1. Теперь добавим на файл SetGID бит с проделаем все то же самое. (Рис. 6, 7).

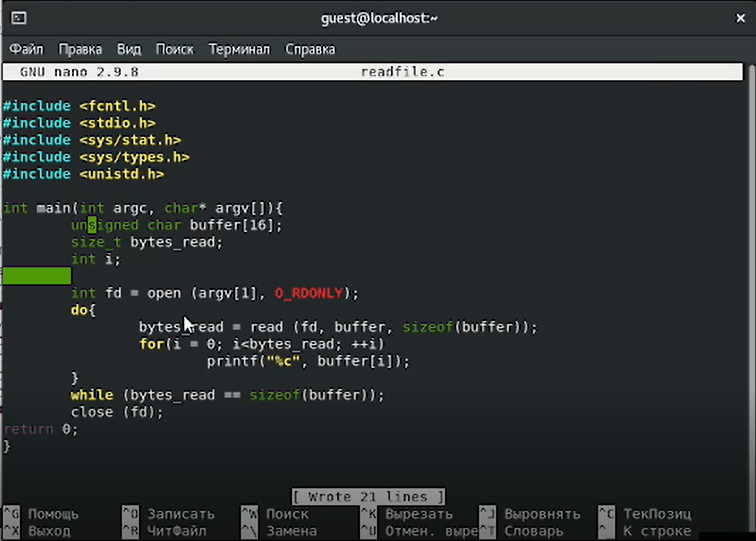


Добавление SetGID бита



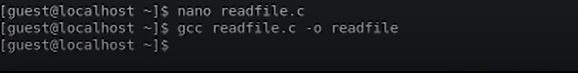
Запуск simpleid2.c

1. Пишем программу readfile.c. (Рис. 8).



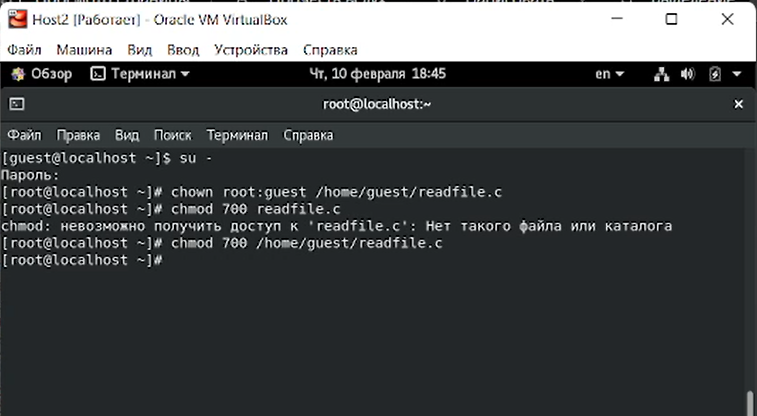
Листинг readfile.c

1. Компилируем программу. (Рис. 9).

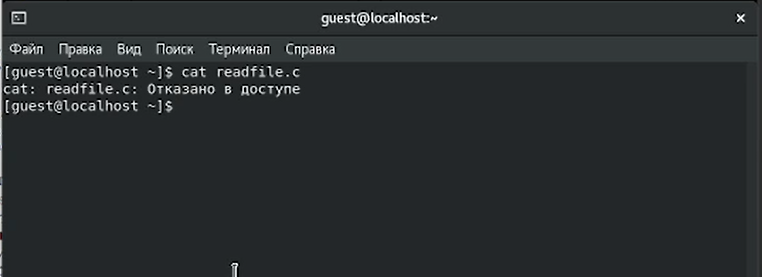


Компиляция

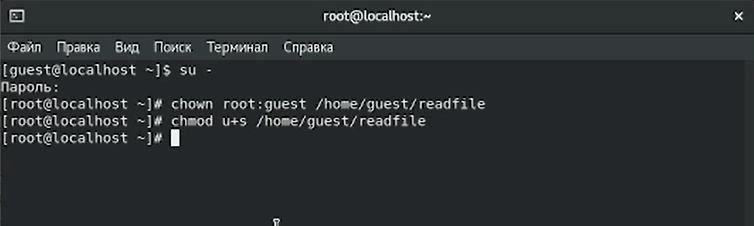
1. Меняем владельца у файла readfile.c и запрещаем чтение всем, кроме суперпользователя. Проверяем, что guest не может читать. Меняем владельца у программы readfile и добавляем SetUID бит на неё. (Рис. 10, 11, 12).



Смена владельца readfile.c

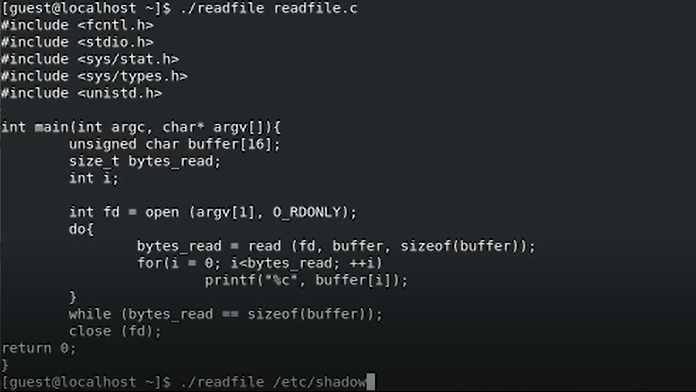
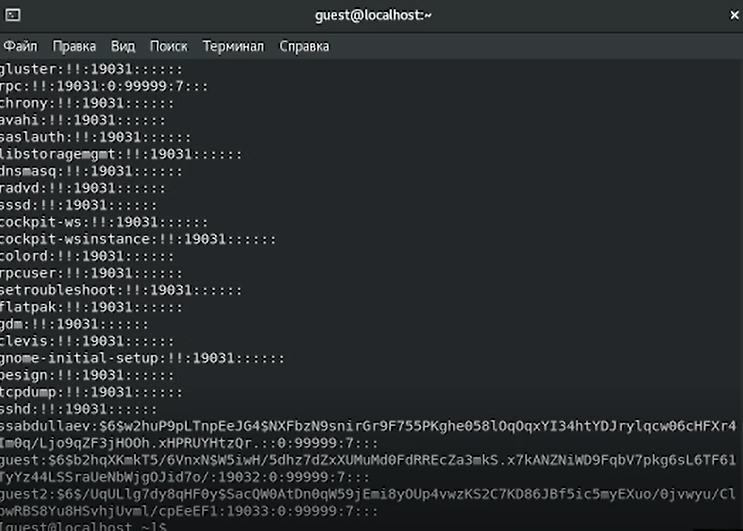
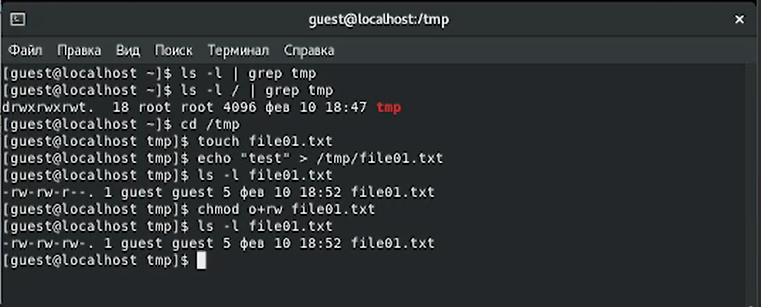
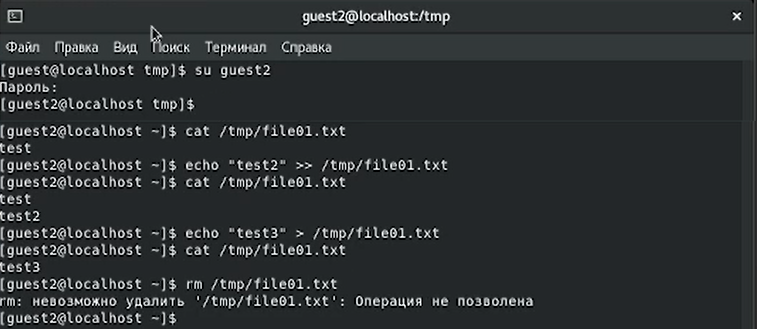
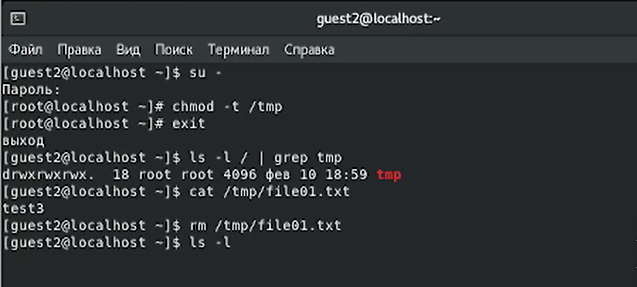
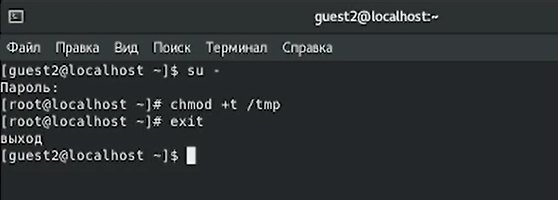


Проверка на cat из-под guest’a



Смена владельца readfile.c

1. Проверяем, может ли программа readfile прочитать файл readfile.c и файл /etc/shadow. Да, может. Хотя сам пользователь вручную не мог. Всё дело в том, что при вызове программы права пользователя повышаются SetUID битом до прав владельца, который может читать файлы (суперпользователь в нашем случае). (Рис. 13, 14).

* 
* Чтение readfile.c программой readfile
* 
* Чтение /etc/shadow программой readfile
  1. Проверяем Sticky бит. Для этого создаем файл, которому даем rw права для others и пишем туда слово test. Теперь пробуем выполнить дозапись в файл, перезапись файла и его удаление. Всё, кроме удаления, прошло успешно. (Рис. 15, 16).
  + 
  + Создание файла, правка прав файла
  + 
  + Проверка прав, тестирование, и попытка удаления
    1. Повышаем права до суперпользователя и удаляем Sticky-бит с папки /tmp. Повторяем наши тесты. Теперь прошли все команды, включая удаление файла. Таким образом, пользователь, не являющийся владельцем файла, смог его удалить, так как Sticky-бит не был настроен. Возвращаем Sticky-бит на папку /tmp. (Рис. 17, 18).
    - 
    - Удаление Sticky-бита и повторное тестирование
    - 
    - Возвращение Sticky-бита

# Выводы

В результате выполнения данной работы были практические навыков работы в консоли с расширенными атрибутами файлов.