# Лабораторнаяработа № 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Адабор Кристофер Твум

13 Арг. 2024

(НКА-бд 03-22)

Российский университет дружбы народов, Москва, Россия

1/17

# Информация

: {. columns align=center}: {. column width= “70%”}

* Адабор Кристофер Твум
* студент Факультета Физико-математических и естесственных наук
* Российский университет дружбы народов
* 1032225824@pfur.ru

Докладчик

https://github.com/Adabor/Lab-3

2/17

# Цельработы

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.

Цель работы

1) Я создал файл “simpleid.c” и внёс в него программу.

Выполнение лабораторной работы

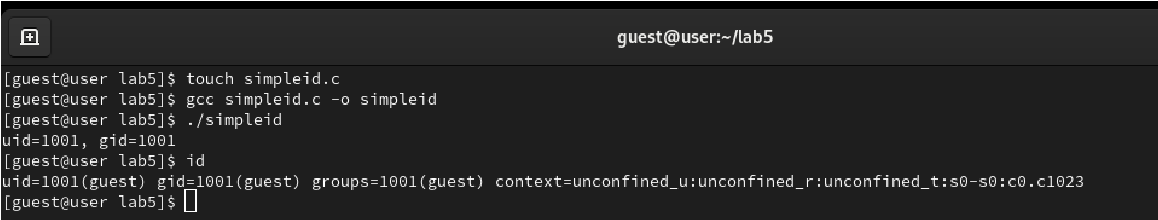
Первая программа 2) Скомпилировал программу и убедился, что файл создан правильно.



Figure 1: Компиляция первой программы

1. Запустил программу и посмотрел, как она работает. Затем прописал команду “id”, чтобы сравнить данные. Все данные сходятся.
2. Создал второй файл и назвал его “simlpeid2.c”. Усложнил первую программу и внёс ее в файл.

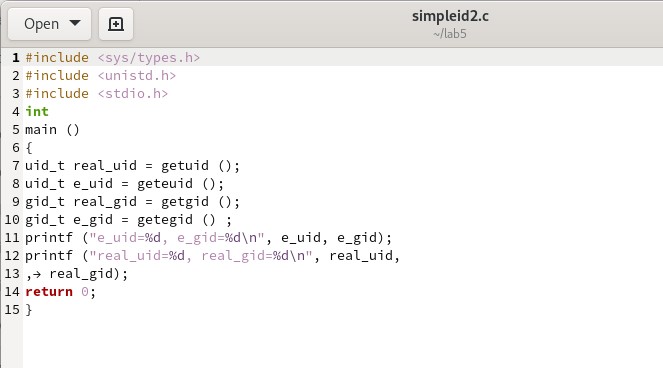


Figure 3: Вторая программа

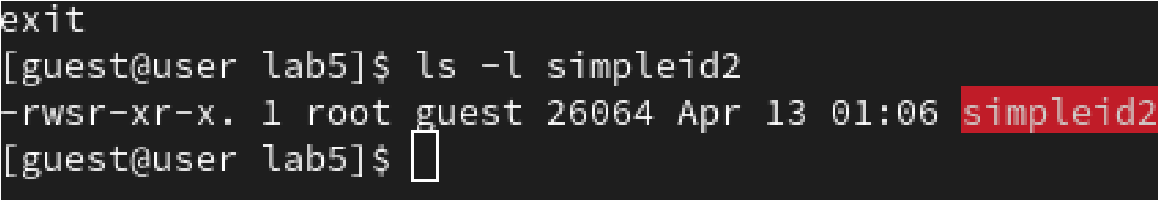
1. Скомпилировал и посмотрел вторую программу. Проверил как она работает.
2. От имени суперпользователя я выполнил команды и временно повысил свои права. Команды сменили пользователя файла на root и установили SetUID-бит. Я запустил файл от имени root-пользователя и проверил сходство с командой “id”.



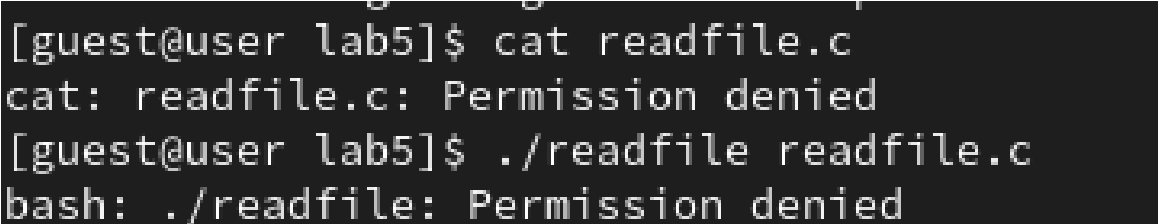
Figure 5: Изменение прав для root



Figure 6: Проверка работы для root

1. Я создал файл “readfile.c”. Внёс туда программу.

Программа readfile

1. Скомпилировал программу readfile.

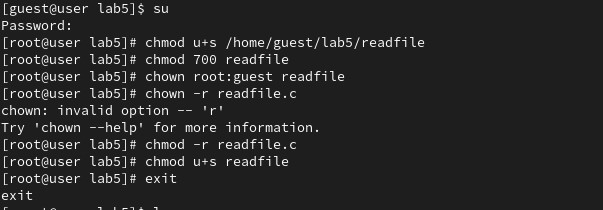
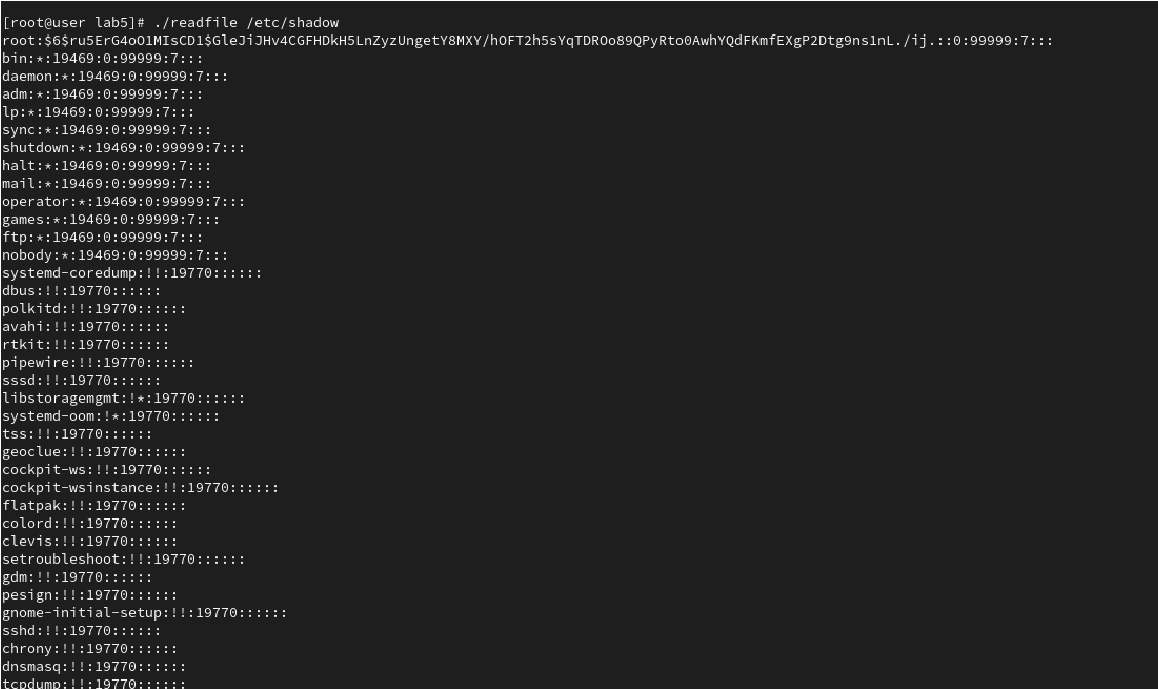


Figure 9: Компиляция readfile

1. Я выдал программе “readfile” права так, чтобы root пользователь мог прочитать файл, а простой пользователь нет.

11) Попытался запустить программу и прочитать два файла с простого пользователя, но программа выдала ошибку. А если запускать с аккаунта root, то программа запускается нормально и работает. Связано это с тем, что владельцем программы является root-пользователь, а у других пользователей нет доступа и прав на использование программы.

1. Я выяснил, установлен ли атрибут Sticky (t) на директории “/tmp”. Атрибут установлен.

Исследование Sticky-бита.



Figure 14: Проверка наличия атрибута

1. От пользователя “guest” я создал файл “file01.txt” в директории “/tmp”. Вписал в файл слово “test”. И дал права на чтение и запись для категории “все остальные (о)”.



Figure 15: Выдача прав для файла

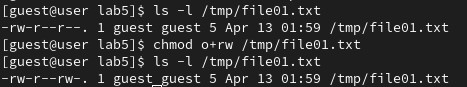


Figure 16: Выдача прав для файла

1. От пользователя “guest2”, который не явлется владельцем, я попробовал прочитать файл. Я могу прочитать файл. Но не могу дописывать содержимое, вписывать новое или удалять этот файл.

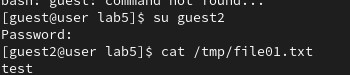


Figure 17: Проверка от второго пользователя

1. я отключил атрибут “t” у директории “/tmp”. Попробовал повторить все предыдущие действия. Я так же не смог вписать в файл данные или дописать их. Но смог прочитать файл и удалить его.



Figure 18: Проверка без атрибута

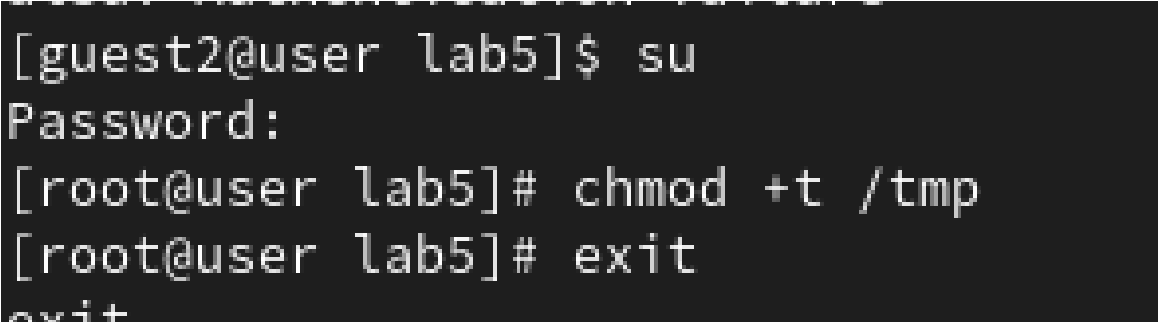
1. Чтобы в дальнейшем у меня не было проблем в работе с директорией “/tmp” я вернул атрибут на директорию, используя суперпользователя.



Figure 19: Возвращение атрибута

Я изучил механизмы изменения идентификатора, применил SetUID-бит и Stickу-бит. Получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работы механизма смены идентификатора процессов пользователя, а так же влияние бита Sticky на запись и удаление файлов.

Выводы

Список литературы