Информационная безопасность. Отчет по лабораторной работе №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Терентьев Егор Дмитриевич 1032192875

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 Выполнение лабораторной работы

Вхожу в систему от guest и создаю программу simpleid.c fig. 1.

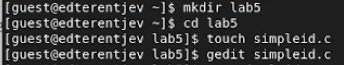


Figure 1: touch\_simpleid

Листинг программы simpleid fig. 2.

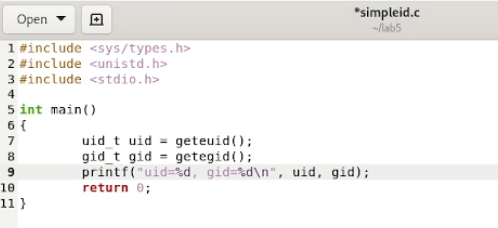


Figure 2: listing\_simpleid

Скомплилируйте программу и убедитесь, что файл программы создан:gcc simpleid.c -o simpleid. Выполните программу simpleid: ./simpleid. Выполните системную программу id: fig. 3.

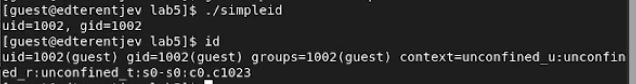


Figure 3: simpleid\_id

Усложните программу, добавив вывод действительных идентификаторов и назовите ее simpleid2 fig. 4

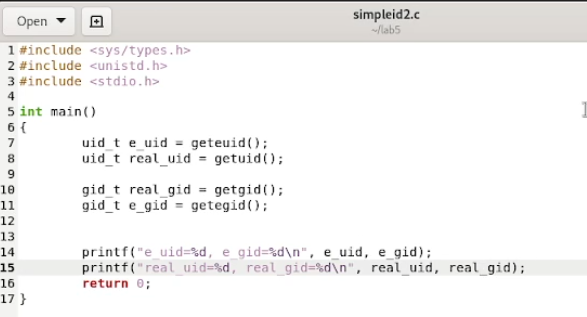


Figure 4: listing\_simpleid2

Скомпилируйте и запустите simpleid2.c fig. 5

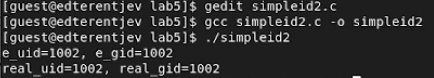


Figure 5: launch\_simpleid

От имени суперпользователя выполните команды: chown root:guest /home/guest/simpleid2 chmod u+s /home/guest/simpleid2 fig. 6.

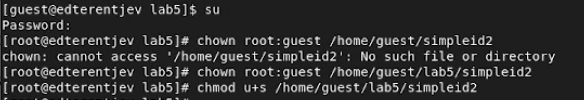


Figure 6: chown\_chmod

Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2: ls -l simpleid2 Запустите simpleid2 и id: ./simpleid2 id fig. 7.

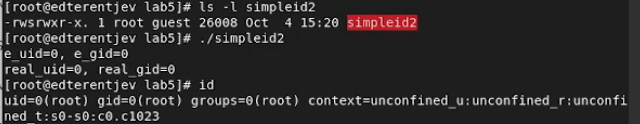


Figure 7: ls\_simpleid2

Проделайте тоже самое относительно SetGID-бита fig. 8

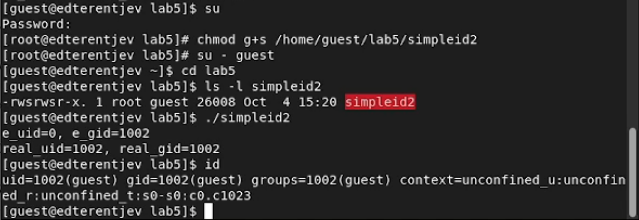


Figure 8: setgid

Создайте программу readfile.c fig. 9.

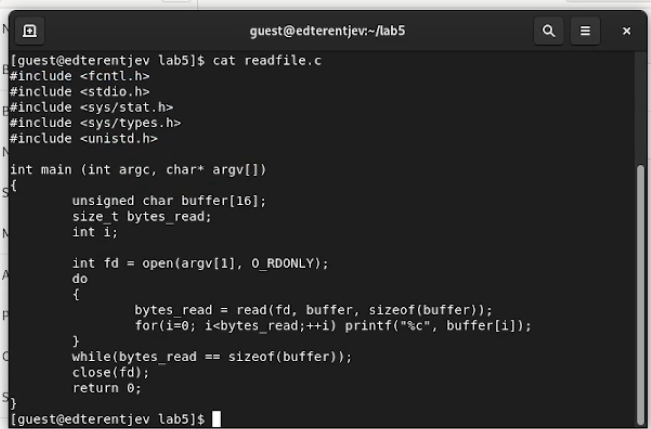


Figure 9: readfile

Смените владельца у файла readfile.c и измените права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог fig. 10.

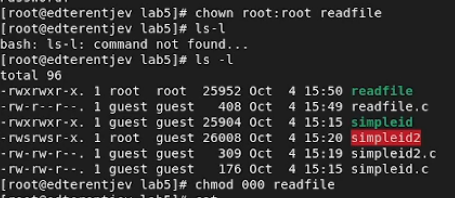


Figure 10: chown\_chmod\_readfile

Проверьте, что пользователь guest не может прочитать файл readfile.c. Смените у программы readfile владельца и установите SetU’D-бит. fig. 11.

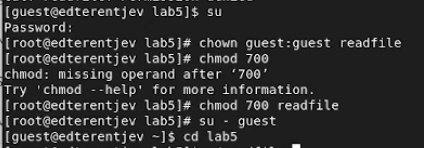


Figure 11: ls\_readfile

Проверьте, может ли программа readfile прочитать файл readfile.c? fig. 12

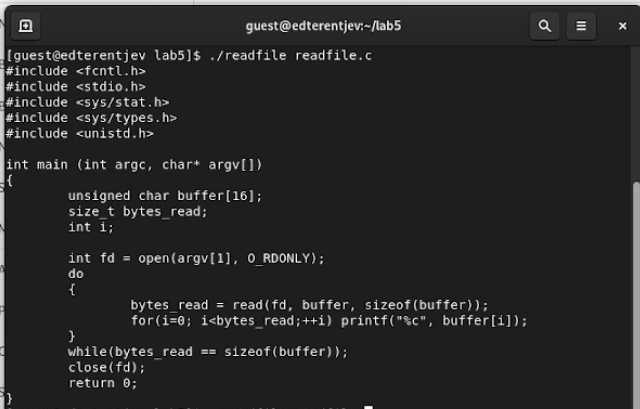


Figure 12: readfile\_readfile

Выясните, установлен ли атрибут Sticky на директории /tmp От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test: Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные» fig. 13

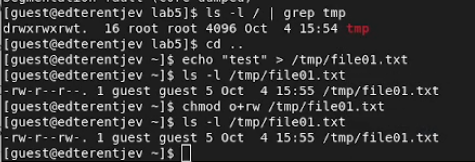


Figure 13: ls\_chmod\_echo

От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 Проверьте содержимое файла От пользователя guest2 попробуйте записать в файл /tmp/file01.txt Проверьте содержимое файла От пользователя guest2 попробуйте удалить файл /tmp/file01.txt fig. 14

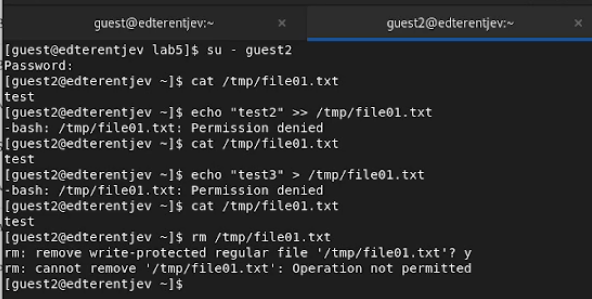


Figure 14: guest2\_comm

Повысьте свои права до суперпользователя и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp От пользователя guest2 проверьте, что атрибута t у директории /tmp нет: Повторите предыдущие шаги fig. 15

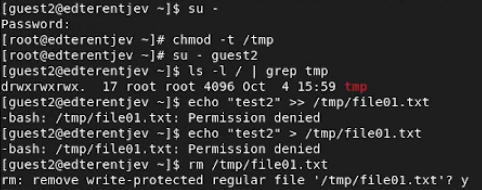


Figure 15: without\_stickybit

Повысьте свои права до суперпользователя и верните атрибут t на директорию /tmp fig. 16

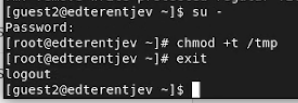


Figure 16: return\_stickybit

# 3 Выводы

В результате выполнения работы я изученил механизмы изменения идентификаторов, применения SetUID- и Sticky-битов, получил практические навыки работы в консоли с дополнительными атрибутами, а также рассмотрел работы механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.

# 4 Список литературы

1. Методические материалы курса