Отчёт по лабораторной работе №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Савченко Елизавета НБИ-01-20

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

## 2.1 Подготовка

1. Для выполнения части заданий требуются средства разработки приложений. Проверили наличие установленного компилятора gcc командой gcc -v: компилятор обнаружен.

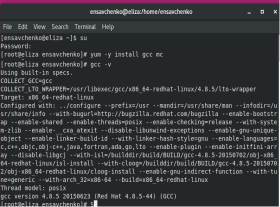


Figure 1: установка компилятора

1. Чтобы система защиты SELinux не мешала выполнению заданий работы, отключили систему запретов до очередной перезагрузки системы командой setenforce 0:
2. Команда getenforce вывела Permissive:

Figure 2: подготовка к работе

Figure 2: подготовка к работе

1. Теперь проверим корректность установки компилятора, как показано на скриншоте номер 3.



Figure 3: проверка установки компилятора

## 2.2 Изучение механики SetUID

1. Вошли в систему от имени пользователя guest.

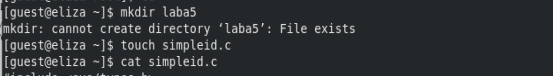


Figure 4: создание и редактирование файла simpleid.c

1. Написали программу simpleid.c.

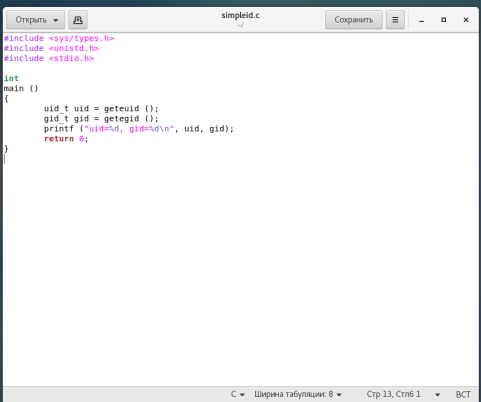


Figure 5: написание программы

1. Скомпилировали программу и убедились, что файл программы создан: gcc simpleid.c -o simpleid
2. Выполнили программу simpleid командой ./simpleid
3. Выполнили системную программу id с помощью команды id. uid и gid совпадает в обеих программах

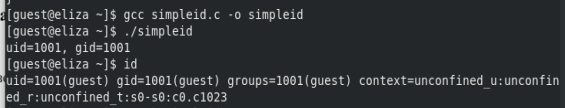


Figure 6: результат программы simpleid

1. Усложнили программу, добавив вывод действительных идентификаторов.

Figure 7: Создание и редактирование файла simpleid2.c

Figure 7: Создание и редактирование файла simpleid2.c

1. Написали следующую программу в файле simpleid2.c

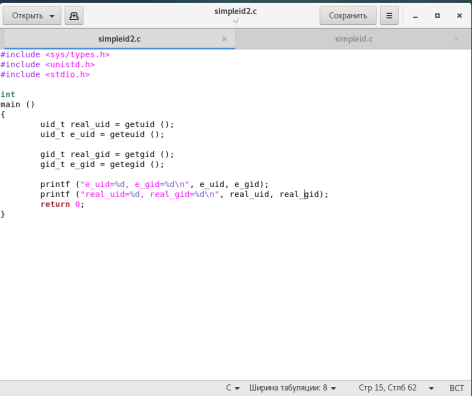


Figure 8: написание программы

1. Скомпилировали и запустили simpleid2.c: gcc

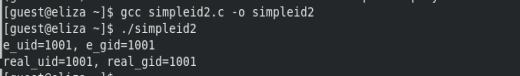


Figure 9: результат программы simpleid2 от guest

1. На скриншоте 10 видно следующее: теперь от имени суперпользователя выполнили команды, которые меняют владельца файла. Использовали su для повыше10 ния прав до суперпользователя. Выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2. Также запустили simpleid2 и id.

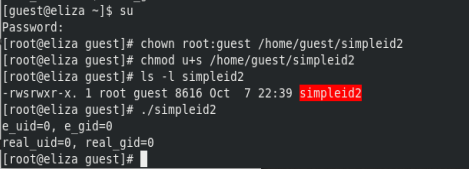


Figure 10: результат программы simpleid2 от root

Приходим к выводу, что результат выполнения программ при заходе от пользователя guest и через root отличается. Теперь создадим файл readfile.c, применяя те же команды, что и в предыдущий раз.

1. Cоздание и редактирование файла

Figure 11: создание и редактирование файла readfile.c

Figure 11: создание и редактирование файла readfile.c

1. Написали программу readfile.c

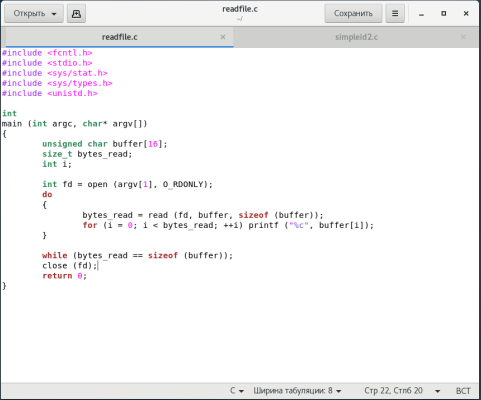


Figure 12: программа readfile

1. Откомпилировали программу. Затем сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог.

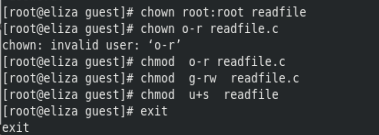


Figure 13: смена прав на файле readfile

1. Проверили, что пользователь guest не может прочитать файл readfile.c. Но увидели, что файл читается.

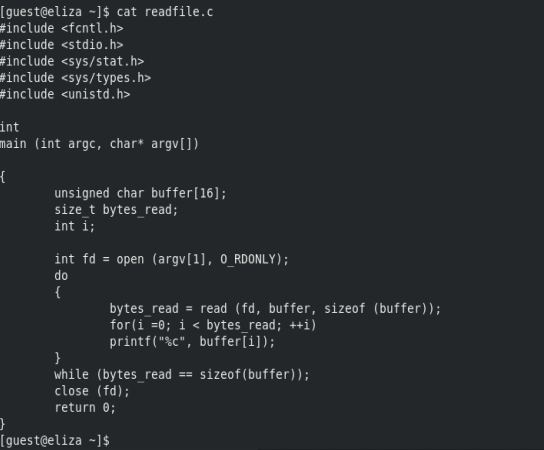


Figure 14: проверка файла на чтение

1. Ищем ошибку. Оказалось, что была допущена ошибка в установке прав. Переделали команду с установкой прав и повторили проверку, на этот раз в доступе отказано, как и должно быть. А команда ./readfile сработала потому что она выполняет чтение от имени суперпользователя, а не guest.

## 2.3 Исследование Sticky-бита

1. От имени пользователя guest создали файл file01.txt в директории /tmp со словом test. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные». Первоначально все группы имели право на чтение, а запись могли осуществлять все, кроме «остальных пользователей».

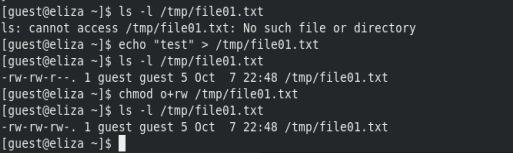


Figure 15: проверка прав на файле file01.txt

1. От пользователя (не являющегося владельцем) попробовали прочитать файл /file01.txt. Мы видим, что в файле записано одно слово. далее к нему мы должны были поочередно дозаписать слова test2 и test3, но я забыла добавить по одной галочке и вместо дозаписи перезаписала слова. От пользователя попробовали удалить файл /tmp/file01.txt командой rm /tmp/file01.txt, однако получила отказ.

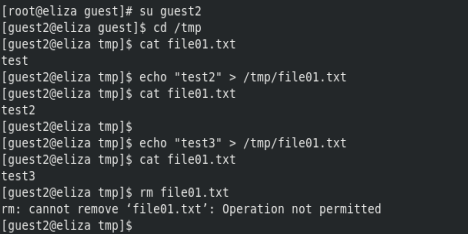


Figure 16: дозапись и перезапись слов в файл file01.txt

1. От суперпользователя командой выполнили команду, снимающую атрибут t (Sticky-бит) с директории /tmp. Покинули режим суперпользователя командой exit. От пользователя проверили, что атрибута t у директории /tmp нет. Повторили предыдущие шаги. Получилось удалить файл.

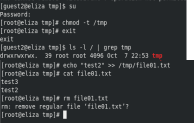


Figure 17: снятие атрибута

# 3 Выводы

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Также мы рассмотрели работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.

# Список литературы

1. [КОМАНДА CHATTR В LINUX](https://losst.ru/neizmenyaemye-fajly-v-linux)
2. [chattr](https://en.wikipedia.org/wiki/Chattr)