Лабораторная работа № 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Тарусов Артём Сергеевич

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

1. Исследовать SetUID- и SetGID-биты.
2. Исследовать Sticky-бит.

# Теоретическое введение

* Операционная система — это комплекс программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем [1].
* Права доступа определяют, какие действия конкретный пользователь может или не может совершать с определенным файлами и каталогами. С помощью разрешений можно создать надежную среду — такую, в которой никто не может поменять содержимое ваших документов или повредить системные файлы. [2].

# Выполнение лабораторной работы

1. От имени пользователя guest создадим программу simpleid.c, скомпилируем ее и убедимся, что файл создан (fig. 1).

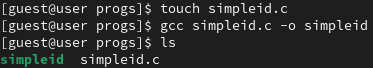


Рис. 1: Создание файла simleid.c

1. Выполним команды ./simpleid и id и убедимся, что полученные данные совпадают (fig. 2).

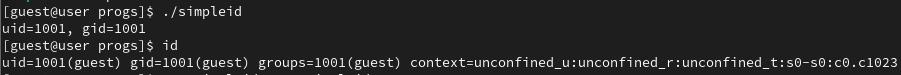


Рис. 2: Использование команд ./simpleid и id

1. Усложним программу и запишем ее в файл simpleid2.c. Запустим получившуюся программу (fig. 3).

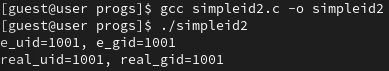


Рис. 3: Создание и запуск программы simpleid2

1. От имени суперпользователя установим новые атрибуты и сменим владельца файла simpleid2 (fig. 4).

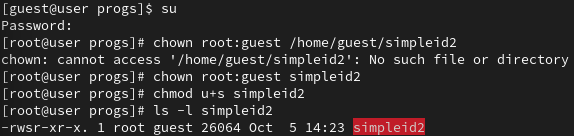


Рис. 4: Установки новых атрибутов и смена владельца файла simpleid2

1. Выполним команды ./simpleid2 и id и убедимся, что полученные данные совпадают (fig. 5).

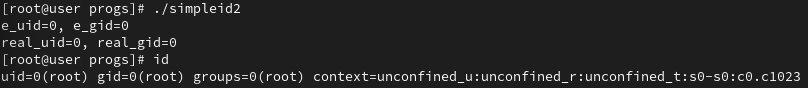


Рис. 5: Использование команд ./simpleid2 и id

1. Проделаем то же самое относительно SetGID-бита (fig. 6).

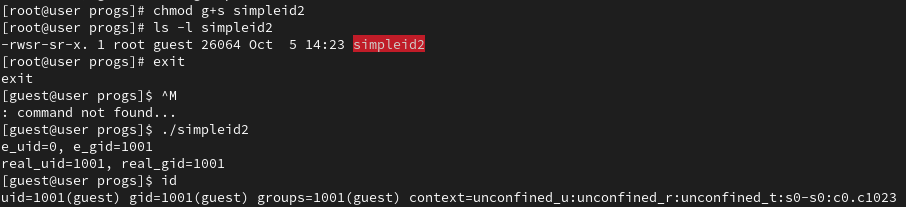


Рис. 6: Операции с SetGID-битом

1. Создадим и скомпилируем программу readfile.c (fig. 7).

Создание и компиляция программы readfile.c

Рис. 7: Создание и компиляция программы readfile.c

1. Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог(fig. 8).

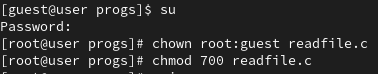


Рис. 8: Изменение владельца и прав файла readfile.c

1. Проверим, что пользователь guest не может прочитать файл readfile.c. (fig. 9).

Проверка, что пользователь guest не может прочитать файл readfile.c.

Рис. 9: Проверка, что пользователь guest не может прочитать файл readfile.c.

1. Сменим у программы readfile владельца и установим SetUID-бит (fig. 10).

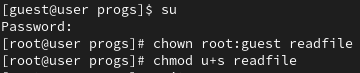


Рис. 10: Работа с параметрами readfile

1. Проверим, может ли программа readfile прочитать файл readfile.c (fig. 11).

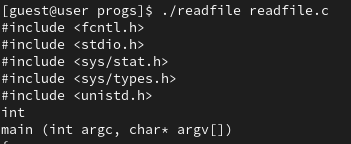


Рис. 11: Попытка прочитать файл readfile.c программой readfile

1. Проверим, может ли программа readfile прочитать файл /etc/shadow (fig. 12).

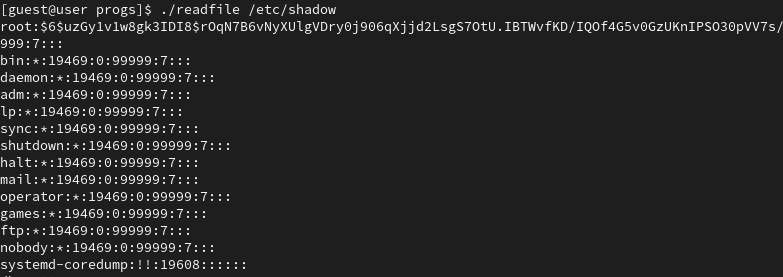


Рис. 12: Попытка прочитать файл /etc/shadow программой readfile

1. Выясним, установлен ли атрибут Sticky на директории /tmp (fig. 13).

Чтение атрибутов директории /tmp

Рис. 13: Чтение атрибутов директории /tmp

1. От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные» (fig. 14).

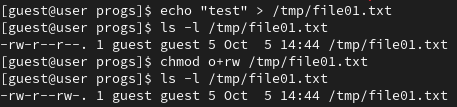


Рис. 14: Чтение атрибутов директории /tmp

1. От пользователя guest2 попробуем прочитать файл /tmp/file01.txt (fig. 15).

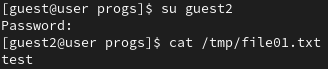


Рис. 15: Попытка прочтения файла /tmp/file01.txt

1. От пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2 (fig. 16).

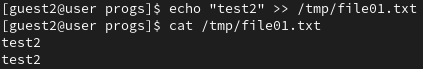


Рис. 16: Попытка дозаписи в файл /tmp/file01.txt

1. От пользователя guest2 попробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию (fig. 17).

Попытка записи в файл /tmp/file01.txt

Рис. 17: Попытка записи в файл /tmp/file01.txt

1. От пользователя guest2 попробуем удалить файл /tmp/file01.txt (fig. 18).

Попытка удаления файла /tmp/file01.txt

Рис. 18: Попытка удаления файла /tmp/file01.txt

1. От имени суперпользователя снимем атрибут t с директории /tmp. От пользователя guest2 проверим, что атрибута t у директории /tmp нет (fig. 19).

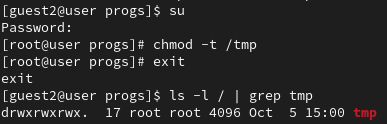


Рис. 19: Удаление атрибута t директории /tmp

1. Повторим предыдущие шаги. Теперь мы можем удалить файл (fig. 20).



Рис. 20: Повторение предыдущих шагов

1. Повысим свои права до суперпользователя и вернем атрибут t на директорию /tmp (fig. 20).

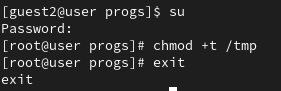


Рис. 21: Возвращение атрибута t директории /tmp

# Выводы

В рамках данной лабораторной работы были изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получены практические навыков работы в консоли с дополнительными атрибутами. Рассмотрены принципы работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов

# Список литературы

[1] https://blog.skillfactory.ru/glossary/operaczionnaya-sistema/

[2] https://codechick.io/tutorials/unix-linux/unix-linux-permissions