Лабораторнаяработа № 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Усов Александр Александрович НБибд-02-18

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 Задание

1. Создать и проверить работу файла simpleid.c
2. Расширить работу файла simpleid2.c и проверить работу с разными атрибутами
3. Создать программу readfile.c и проверить атрибуты на нем.

# 3 Теоретическое введение

setuid и setgid являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами владельца или группы исполняемого файла. В Unix-подобных системах приложение запускается с правами пользователя, вызвавшего указанное приложение. Это обеспечивает дополнительную безопасность, так как процесс с правами пользователя не сможет получить доступ на запись к важным системным файлам, например /etc/passwd, который принадлежит суперпользователю root.

Более подробно см. в [1].

Sticky bit — дополнительный атрибут файлов или каталогов в операционных системах семейства UNIX. Впервые sticky bit появился в пятой редакции UNIX в 1974 году для использования в исполняемых файлах. Он применялся для уменьшения времени загрузки наиболее часто используемых программ. После закрытия программы код и данные оставались в памяти, а следующий запуск происходил быстрее

Более подробно см. в [2].

# 4 Выполнение лабораторной работы

## 4.1 Создание программы

1. Вошли в систему от имени пользователя guest.
2. Создали программу simpleid.c. Для этого запустили редактор vi с помощью команды vi simpleid.c:

(рис. 1):

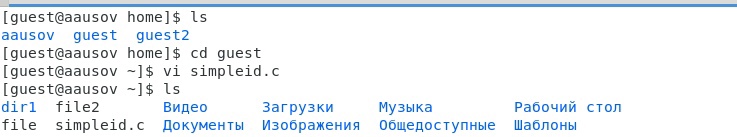


Figure 1: Создание программы

Напечатали текст программы в открывшемся редакторе:(рис. 2).



Figure 2: Программа

1. Скомпилировали программу и убедилаись, что файл программы создав: gcc simpleid.c -o simpleid

(рис. 3).

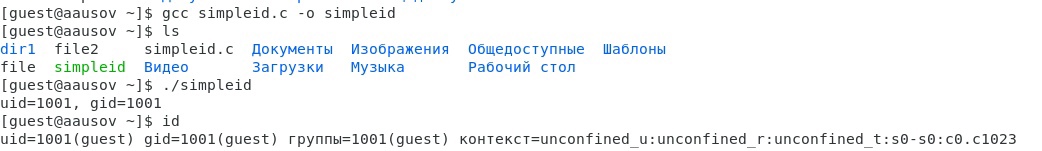


Figure 3: Команда chmod 600 file1

1. Выполнили программу simpleid командой ./simpleid

(рис. 3).

1. Выполнили системную программу id с помощью команды id (рис. 3). uid и gid совпадает в обеих программах
2. Усложнили программу, добавив вывод действительных идентификаторов. Получившуюся программу назвали simpleid2.c. (рис. 4).

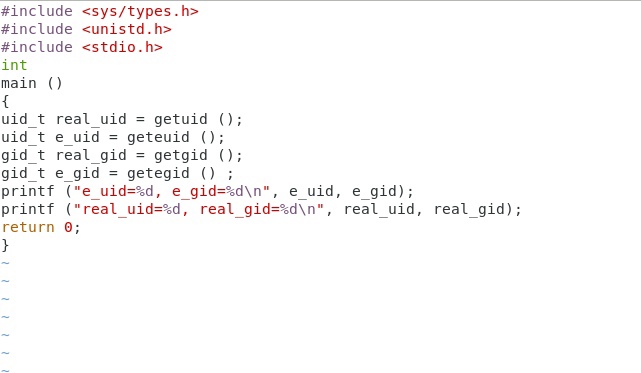


Figure 4: Усложненная программа

1. Скомпилировали и запустили simpleid2.c: gcc simpleid2.c -o simpleid2 ./simpleid2

(рис. 5)

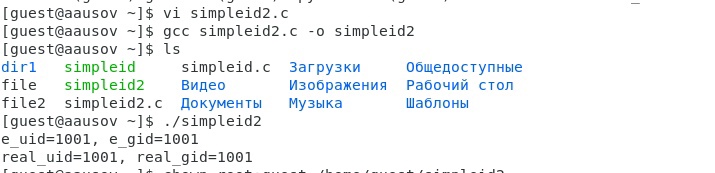


Figure 5: Скомпилировали и запустили

1. От имени суперпользователя выполнили команды: chown root:guest /home/guest/simpleid2 chmod u+s /home/guest/simpleid2

(рис. 6)

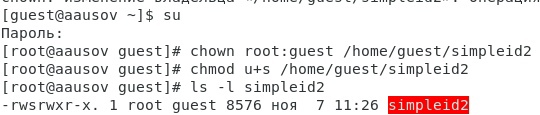


Figure 6: Выполнили команды

1. Использовали su для повышения прав до суперпользователя
2. Выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2: ls -l simpleid2 (рис. 6)
3. Запустили simpleid2 и id: ./simpleid2, id Результат выполнения программ теперь немного отличается (рис. 7)

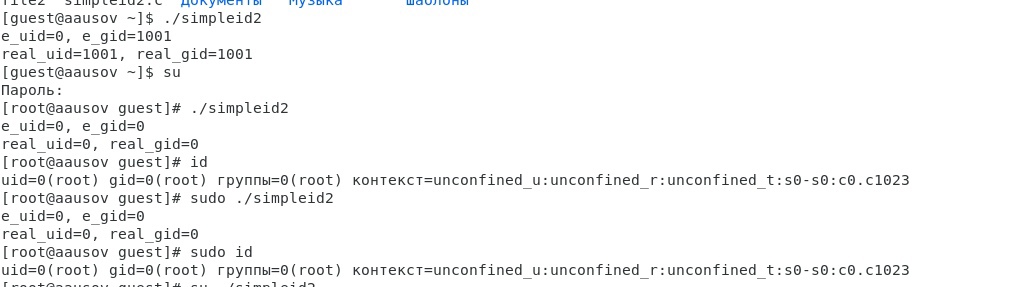


Figure 7: Перезапустили

1. Проделали тоже самое относительно SetGID-бита. От имени суперпользователя выполним команду chmod g+s /home/guest/simpleid2 Запустили simpleid2 и id (рис. 8)



Figure 8: chmod g+s /home/guest/simpleid2

1. Создали программу readfile.c с помощью команды vi readfile.c (рис. 9)

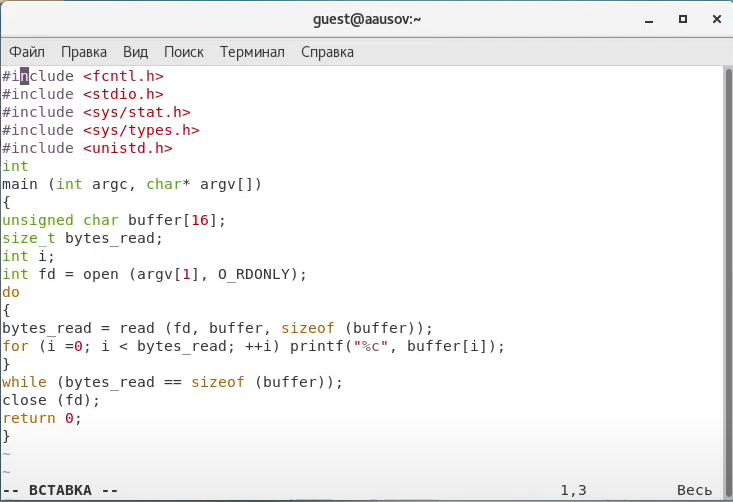


Figure 9: readfile.c

1. Откомпилировали её. (рис. 10) gcc readfile.c -o readfile

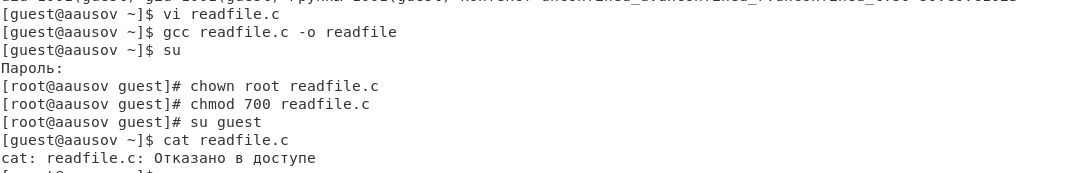


Figure 10: gcc readfile.c -o readfile

1. Сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог. chown root:guest /home/guest/readfile.c chmod 700 /home/guest/readfile.c (рис. 10)
2. Проверили, что пользователь guest не может прочитать файл readfile.c. (рис. 10)
3. Сменили у программы readfile владельца и установили SetU’D-бит. (рис. 11)
4. Проверили, может ли программа readfile прочитать файл readfile.c (рис. 11)
5. Проверили, может ли программа readfile прочитать файл /etc/shadow (рис. 11)



Figure 11: может ли программа readfile прочитать файл readfile.c

## 4.2 Исследование Sticky-бита

1. Выяснили, установлен ли атрибут Sticky на директории /tmp: ls -l / | grep tmp (рис. 12)

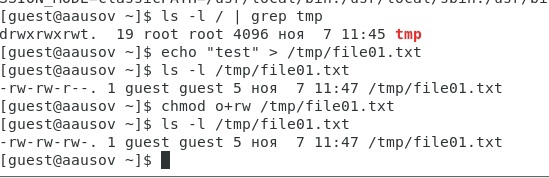


Figure 12: Выяснили, установлен ли атрибут Sticky на директории

1. От имени пользователя guest создали файл file01.txt в директории /tmp со словом test: echo “test” > /tmp/file01.txt (рис. 12)
2. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные»: ls -l /tmp/file01.txt chmod o+rw /tmp/file01.txt ls -l /tmp/file01.txt (рис. 12) Первоначально все группы имели право на чтение, а запись могли осуществлять все, кроме «остальных пользователей».
3. От пользователя guest2 прочитать файл /tmp/file01.txt: cat /tmp/file01.txt (рис. 13)

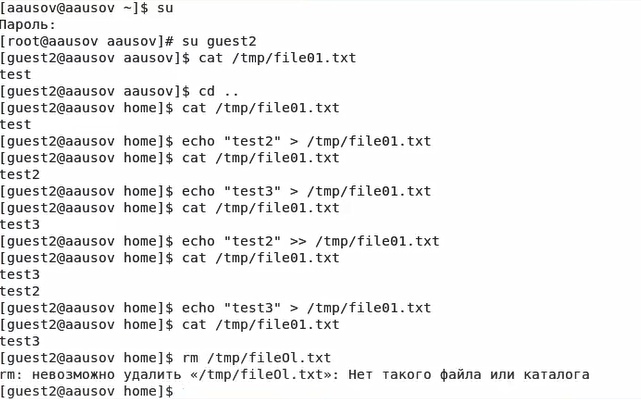


Figure 13: Действи с файлом

1. От пользователя guest2 попробовали дозаписать в файл /tmp/file01.txt слово test3 командой: echo “test2” >> /tmp/file01.txt (рис. 13)
2. Проверили содержимое файла командой: cat /tmp/file01.txt (рис. 13)
3. От пользователя guest2 попробовали записать в файл /tmp/file01.txt слово test4, стерев при этом всю имеющуюся в файле информацию командой. Для этого воспользовались командой echo “test3” > /tmp/file01.txt Проверили содержимое файла командой (рис. 13)
4. От пользователя guest2 попробовал удалить файл /tmp/file01.txt командой rm /tmp/file01.txt, однако получил отказ. (рис. 13)
5. От суперпользователя командой выполнили команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp (рис. 14)

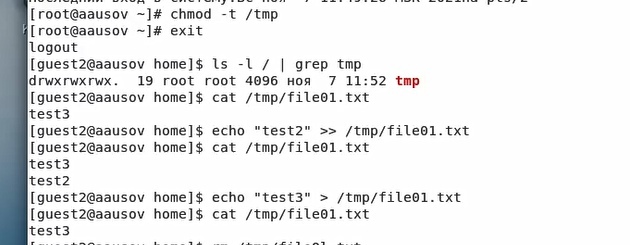


Figure 14: Снял атрибут t

1. От пользователя guest2 проверили, что атрибута t у директории /tmp нет: ls -l / | grep tmp (рис. 14)
2. Повторили предыдущие шаги. (рис. 14) Получилось удалить файл (рис. 15)

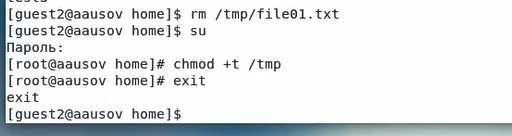


Figure 15: Удалил файл

1. Повысили свои права до суперпользователя и вернули атрибут t на директорию /tmp : su chmod +t /tmp exit (рис. 15)

# 5 Выводы

Изучил механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получил практические навыки работы в консоли с дополнительными атрибутами. Также я рассмотрел работы механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.

# Список литературы

1. suid [Электронный ресурс]. Сайт, 2021. URL: <https://ru.wikipedia.org/wiki/Suid>.

2. Sticky bit [Электронный ресурс]. Сайт, 2021. URL: <https://ru.wikipedia.org/wiki/Sticky_bit>.