Отчет по лабораторной работе №5

-

Овениязов Артур

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

Лабораторная работа подразумевает практическое исследование дискреционных разграничений в современных системах с открытым кодом на базе ОС Linux, а именно изучение атрибутов для групп пользователей.

# 3 Теоретическое введение

В Linux, как и в любой многопользовательской системе, абсолютно естественным образом возникает задача разграничения доступа субъектов — пользователей к объектам — файлам дерева каталогов. Один из подходов к разграничению доступа — так называемый дискреционный (от англ, discretion — чье-либо усмотрение) — предполагает назначение владельцев объектов, которые по собственному усмотрению определяют права доступа субъектов (других пользователей) к объектам (файлам), которыми владеют. Дискреционные механизмы разграничения доступа используются для разграничения прав доступа процессов как обычных пользователей, так и для ограничения прав системных программ в (например, служб операционной системы), которые работают от лица псевдопользовательских учетных записей. [1]

# 4 Выполнение лабораторной работы

## 4.1 ВАЖНО!!! Пунктов в отчете было бы необычайного много если отвечать скрином на каждый, поэтому многие пункты были объеденены для более легкого составления отчета .

## 4.2 Часть первая

Войдите в систему от имени пользователя guest.Создайте программу simpleid.c. Скомплилируйте программу и убедитесь, что файл программы создан:gcc simpleid.c -o simpleid Выполните программу simpleid:./simpleid Выполните системную программу id: id и сравните полученный вами результат с данными предыдущего пункта задания. Программа и команда выдают одинаковые результаты.

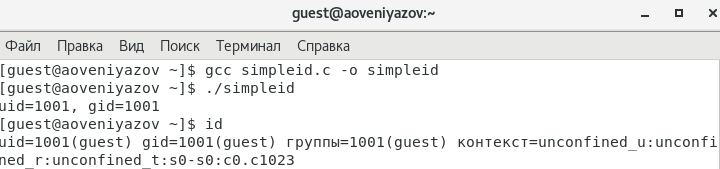


Figure 1: Результат компилирования simpleid и id

Усложните программу, добавив вывод действительных идентификаторов,cкомпилируйте и запустите simpleid2.c: gcc simpleid2.c -o simpleid2 ./simpleid2

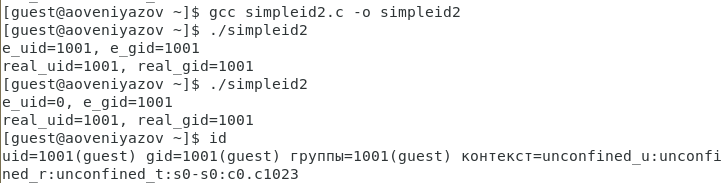


Figure 2: Результат компилирования simpleid2

От имени суперпользователя выполните команды chown root:guest /home/guest/simpleid2 chmod u+s /home/guest/simpleid2 Chmod u+s наделяет каждого пользователя правами владельца(кто пытается получить к нему доступ), а chown меняет владельца. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2: ls -l simpleid2 Запустите simpleid2 и id: ./simpleid2 id Результаты соврадают.

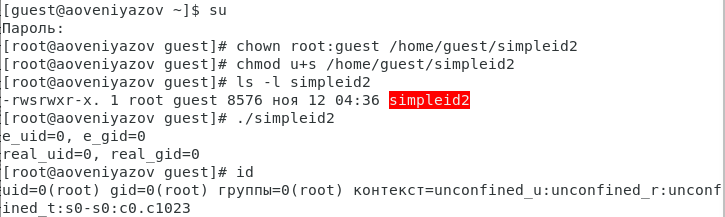


Figure 3: Результат chmod, chown, simpleid2, id

Создайте программу readfile.c.Откомпилируйте её. gcc readfile.c -o readfile

Figure 4: Результат компилирования readfile.c

Figure 4: Результат компилирования readfile.c

Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог

Figure 5: Смена прав и владельца readfile.c

Figure 5: Смена прав и владельца readfile.c

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

Figure 6: Проверка чтения файла

Figure 6: Проверка чтения файла

Смените у программы readfile владельца и установите SetU’D-бит

Figure 7: Обратная смена владельца файла

Figure 7: Обратная смена владельца файла

Проверьте, может ли программа readfile прочитать файл readfile.c? Проверьте, может ли программа readfile прочитать файл /etc/shadow? Чтение обоих файлов не удалось.

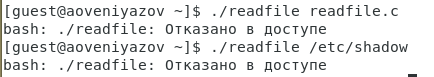


Figure 8: Проверка чтения

## 4.3 Часть вторая

Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду ls -l / | grep tmp. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test: echo “test” > /tmp/file01.txt.Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»: ls -l /tmp/file01.txt chmod o+rw /tmp/file01.txt ls -l /tmp/file01.txt

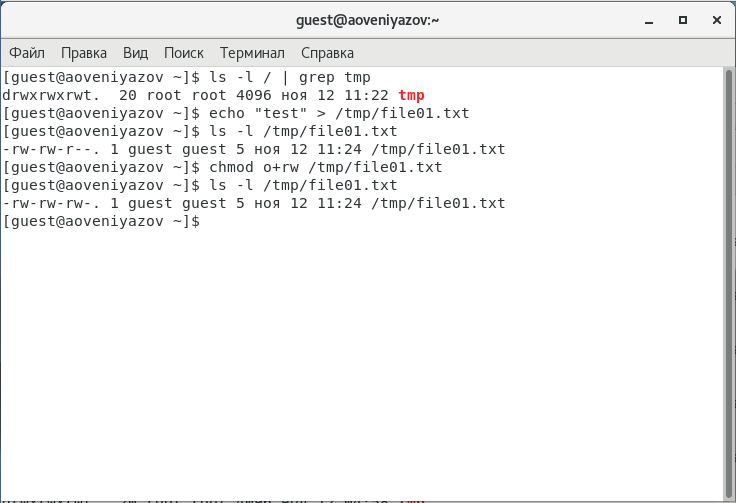


Figure 9: Результат chmod и ls -l /tmp/file01.txt

От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt: cat /tmp/file01.txt5., попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой echo “test2” > /tmp/file01.txt. Проверьте содержимое файла командой cat /tmp/file01.txt От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo “test3” > /tmp/file01.txt Проверьте содержимое файла командой cat /tmp/file01.txt От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой rm /tmp/fileOl.txt Все операции были выполнены успешно.



Figure 10: Результат cat и echo

Повысьте свои права до суперпользователя следующей командой su - и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp. Покиньте режим суперпользователя командой exit. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет: ls -l / | grep tmp. Повторите предыдущие шаги. Какие наблюдаются изменения? Ваши наблюдения занесите в отчёт - все удалось.

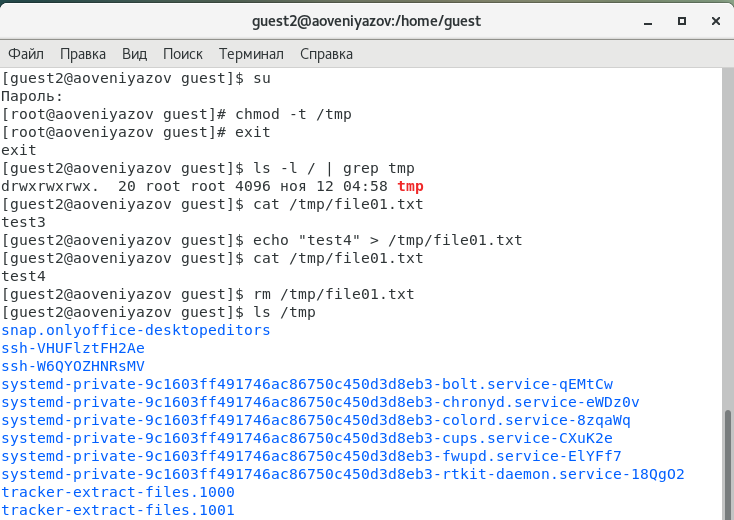


Figure 11: Результат chmod -t /tmp

Повысьте свои права до суперпользователя и верните атрибут t на директорию /tmp: su - chmod +t /tmp exit

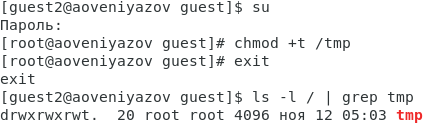


Figure 12: Результат su/chmod +t

# 5 Выводы

Сегодня я приобрел практические навыки работы в консоли с атрибутами файлов, закрепил теоретические основы дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux и изучил механизмы изменения идентификаторов применения SetUID- и Sticky-битов.

# Список литературы

1. Дискреционное разграничение доступа Linux [Электронный ресурс]. Сайт, 2021. URL: <http://debianinstall.ru/diskretsionnoe-razgranichenie-dostupa-linux/>.