Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Липатникова М.С. группа НФИбд-02-19

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

Удостоверилась, что установлен gcc (fig. 1). Вошла в систему от имени пользователя guest. Создала программу simpleid.c:

#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
int  
main ()  
  
{  
  
uid\_t uid = geteuid ();  
  
gid\_t gid = getegid ();  
  
printf ("uid=%d, gid=%d\n", uid, gid);  
  
return 0;  
  
}

Скомплилировала программу и убедилась, что файл программы создан: gcc simpleid.c -o simpleid. Выполнила программу simpleid: ./simpleid. Выполнила системную программу id: id. Полученный результат с данными предыдущего пункта задания совпадает (fig. 2).

Figure 1: Проверка gcc

Figure 1: Проверка gcc



Figure 2: Работа с программой simpleid

Усложнила программу, добавив вывод действительных идентификаторов:

#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
int  
main ()  
  
{  
  
uid\_t real\_uid = getuid ();  
  
uid\_t e\_uid = geteuid ();  
  
gid\_t real\_gid = getgid ();  
  
gid\_t e\_gid = getegid () ;  
  
printf ("e\_uid=%d, e\_gid=%d\n", e\_uid, e\_gid);  
  
printf ("real\_uid=%d, real\_gid=%d\n", real\_uid,  
  
real\_gid);  
  
return 0;  
  
}

Получившуюся программу назвала simpleid2.c. Скомпилировала и запустила simpleid2.c (fig. 3):

gcc simpleid2.c -o simpleid2

./simpleid2

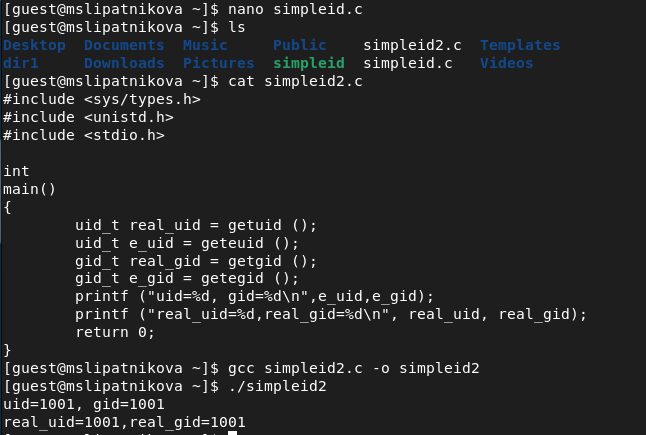


Figure 3: Работа с программой simpleid2

От имени суперпользователя выполнила команды (fig. 6):

chown root:guest /home/guest/simpleid2

chmod u+s /home/guest/simpleid2

Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2: ls -l simpleid2. Запустила simpleid2 и id:

./simpleid2

id

Замечаем, что все совпадает, кроме uid = 0 (fig. 4).

Проделала тоже самое относительно SetGID-бита (fig. 5).

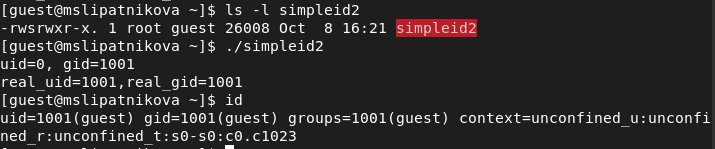


Figure 4: Работа simpleid2(u+s)

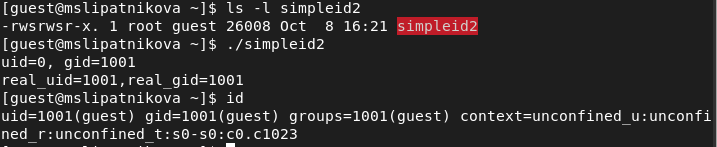


Figure 5: Работа simpleid2(g+s)

Figure 6: Команды от суперпользователя

Figure 6: Команды от суперпользователя

Создала программу readfile.c:

#include <fcntl.h>  
#include <stdio.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
  
int  
main (int argc, char\* argv[])  
  
 {  
  
 unsigned char buffer[16];  
  
 size\_t bytes\_read;  
  
 int i;  
  
 int fd = open (argv[1], O\_RDONLY);  
  
 do  
  
 {  
  
 bytes\_read = read (fd, buffer, sizeof (buffer));  
  
 for (i =0; i < bytes\_read; ++i) printf("%c", buffer[i]);  
  
 }  
  
 while (bytes\_read == sizeof (buffer));  
  
 close (fd);  
  
 return 0;  
  
 }

Откомпилировала её: gcc readfile.c -o readfile (fig. 7). Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог (fig. 9). Проверила, что пользователь guest не может прочитать файл readfile.c (fig. 8). Сменила у программы readfile владельца и установила SetU’D-бит (fig. 9). Проверила, может ли программа readfile прочитать файл readfile.c (может) (fig. 10). Проверила, может ли программа readfile прочитать файл /etc/shadow (может) (fig. 10).

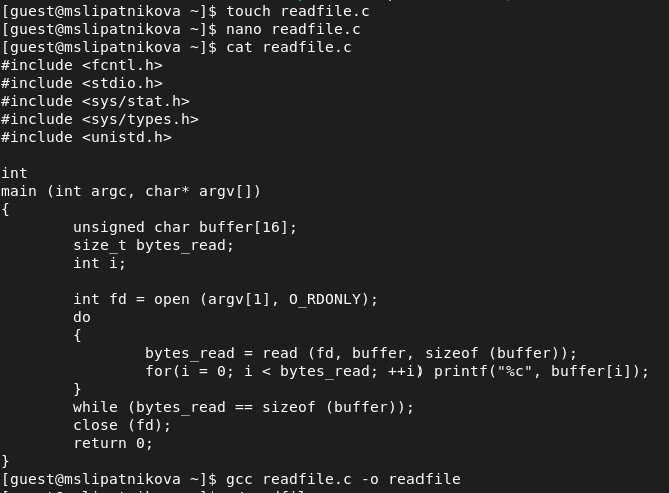


Figure 7: Программа readfile

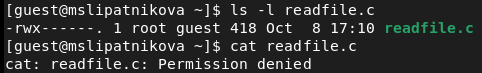


Figure 8: Чтение readfile.c

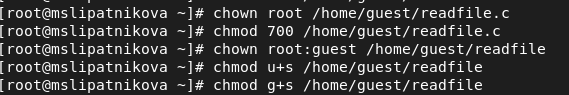


Figure 9: Команды от суперпользователя

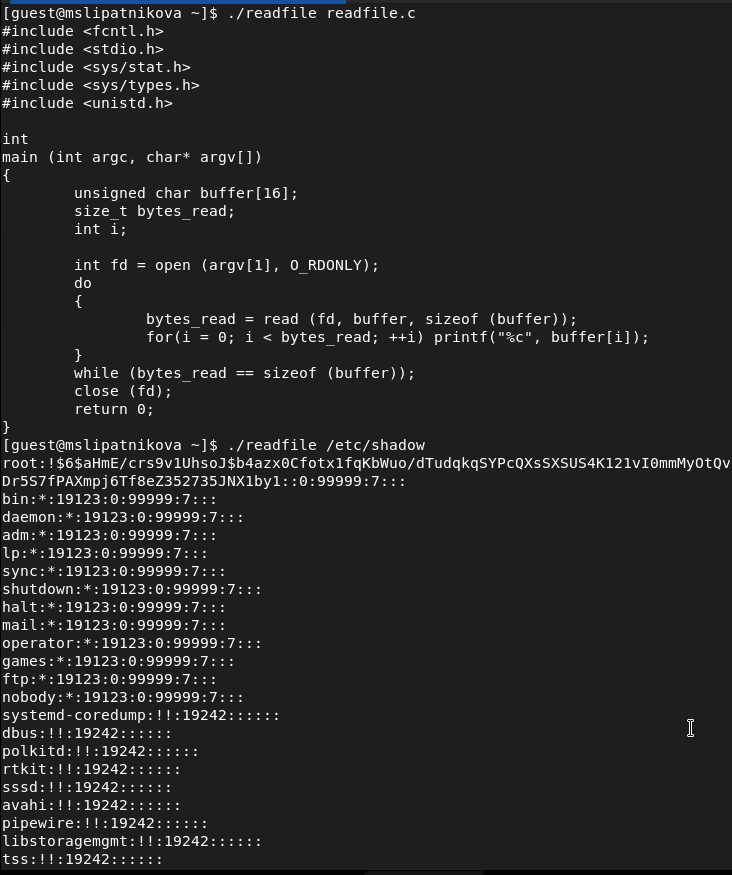


Figure 10: Чтение с помощью программы readfile

Выяснила, установлен ли атрибут Sticky на директории /tmp, для чего выполнила команду: ls -l / | grep tmp. От имени пользователя guest создала файл file01.txt в директории /tmp со словом test: echo “test” > /tmp/file01.txt. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные»:

ls -l /tmp/file01.txt

chmod o+rw /tmp/file01.txt

ls -l /tmp/file01.txt

От пользователя guest2 (не являющегося владельцем) попробовала прочитать файл /tmp/file01.txt: cat /tmp/file01.txt. От пользователя guest2 попробовала дозаписать в файл /tmp/file01.txt слово test2 командой: echo “test2” >> /tmp/file01.txt. Удалось выполнить операцию. Проверила содержимое файла командой: cat /tmp/file01.txt. От пользователя guest2 попробовала записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой: echo “test3” > /tmp/file01.txt. Удалось выполнить операцию. Проверила содержимое файла командой: cat /tmp/file01.txt. От пользователя guest2 попробовала удалить файл /tmp/file01.txt командой: rm /tmp/fileOl.txt. Не удалось удалить файл (fig. 11).

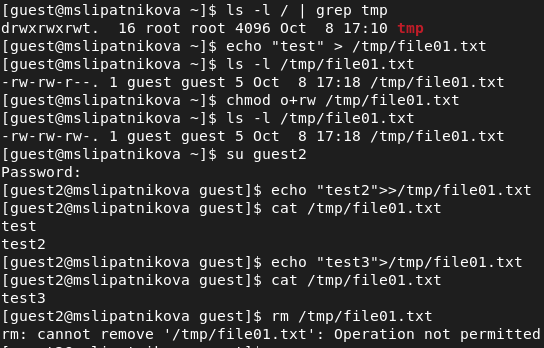


Figure 11: Работа с tmp/file01.txt

От суперпользователя выполнила команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp (fig. 13). От пользователя guest2 проверила, что атрибута t у директории /tmp нет: ls -l / | grep tmp. Повторила предыдущие шаги. Удалось в этот раз удалить файл от имени пользователя, не являющегося его владельцем (fig. 12). От суперпользователя выполнила команду, вернувший атрибут t (Sticky-бит) в директории /tmp: chmod +t /tmp (fig. 13).

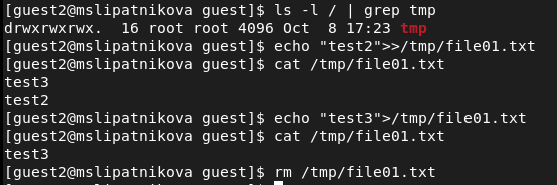


Figure 12: Работа с tmp/file01.txt без t

Figure 13: Команды от суперпользователя

Figure 13: Команды от суперпользователя

# 3 Вывод

Изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

# 4 Список литературы

1. Теоретические материалы курса.