Лабораторная работа № 5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Дмитревская Софья Алексеевна, НФИбд-01-19

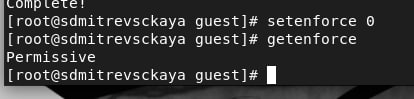
Содержание

# 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 Последовательность выполнения работы

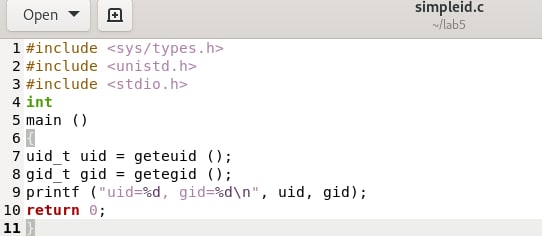
## 2.1 Создание программы

1. Войдите в систему от имени пользователя guest. 
2. 

* Figure 1: создание и редактура файла

1. Создайте 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;  
   }

1. 

* Figure 2: Прогамма

1. Скомплилируйте программу и убедитесь, что файл программы создан:

* gcc simpleid.c -o simpleid

1. Выполните программу simpleid:

* ./simpleid

1. Выполните системную программу id:

* id

и сравните полученный вами результат с данными предыдущего пункта задания.

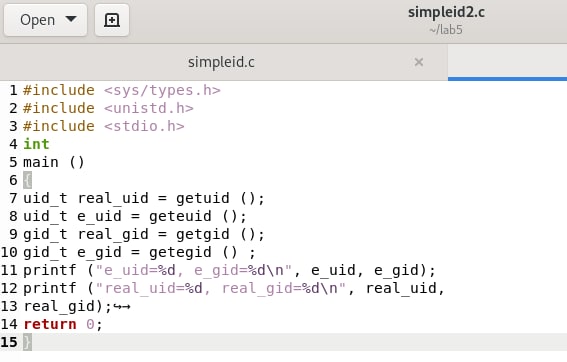


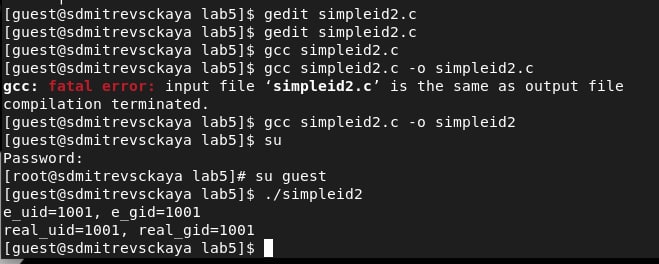
Figure 3: компиляция и запуск

1. Усложните программу, добавив вывод действительных идентификато- ров:

* #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 назовите simpleid2.c.

1. Усложните программу 
2. Скомпилируйте и запустите simpleid2.c:

* gcc simpleid2.c -o simpleid2  
  ./simpleid2
* 
* Figure 4: simpleid2

1. Создайте 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;  
  }

1. 

* Figure 5: readfile

1. Откомпилируйте её.

* gcc readfile.c -o readfile

Figure 6: проверка возможности чтения файла

Отразите полученный результат и ваши объяснения в отчёте.

## 2.2 Исследование Sticky-бита

1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду

* ls -l / | grep tmp

1. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test:

* echo "test" > /tmp/file01.txt

1. Просмотрите атрибуты у только что созданного файла и разрешите чте- ние и запись для категории пользователей «все остальные»:

* ls -l /tmp/file01.txt  
   chmod o+rw /tmp/file01.txt  
   ls -l /tmp/file01.txt

1. От пользователя guest2 (не являющегося владельцем) попробуйте про- читать файл /tmp/file01.txt:

* cat /tmp/file01.txt

1. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой

* echo "test2" > /tmp/file01.txt

Удалось ли вам выполнить операцию?

1. Проверьте содержимое файла командой

* cat /tmp/file01.txt

1. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию ко- мандой

* echo "test3" > /tmp/file01.txt

Удалось ли вам выполнить операцию?

1. Проверьте содержимое файла командой

* cat /tmp/file01.txt

1. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt ко- мандой

* rm /tmp/fileOl.txt

1. Повысьте свои права до суперпользователя следующей командой

* su -

и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp:

chmod -t /tmp

1. Покиньте режим суперпользователя командой exit
2. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет:

* ls -l / | grep tmp

1. Скриншот выполнения:

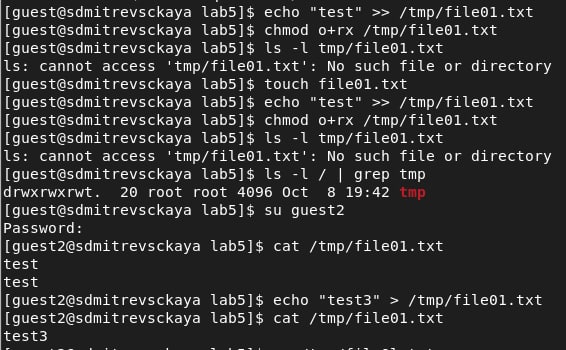


Figure 7: Скриншот выполнения

# 3 Выводы

Изучены механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получены практическе навыки работы в консоли с дополнительными атрибутами. Рассмотрены работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

# 4 Библиография

1. Методические материалы курса