Отчет по лабораторной работе №5

по дисциплине: Информационная безопасность

Ким Михаил Алексеевич

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. Произвести эксперименты с дополнительными атрибутами прав доступа.

# 3 Теоретическое введение

## 3.1 Термины

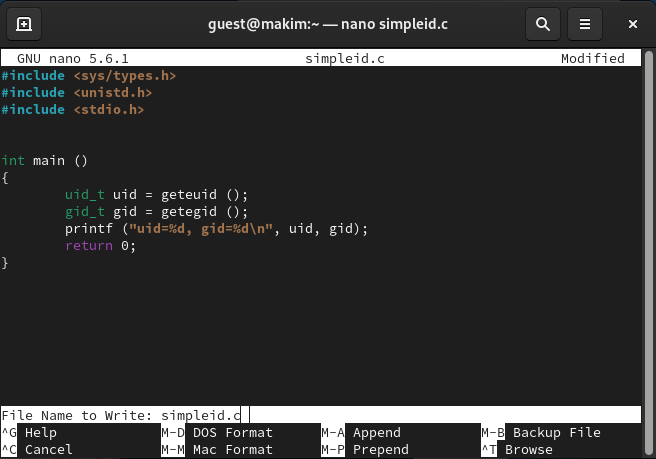
* Терминал (или «Bash», сокращение от «Bourne-Again shell») — это программа, которая используется для взаимодействия с командной оболочкой. Терминал применяется для выполнения административных задач, например: установку пакетов, действия с файлами и управление пользователями. [1]
* Права доступа определяют, какие действия конкретный пользователь может или не может совершать с определенным файлами и каталогами. [2]
* В UNIX-системах, кроме стандартных прав доступа, существуют также дополнительные или специальные атрибуты файлов, которые поддерживает файловая система. [3]
* Setuid – это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо.
* Sticky Bit - в случае, если этот бит установлен для папки, то файлы в этой папке могут быть удалены только их владельцем. [4]

# 4 Выполнение лабораторной работы

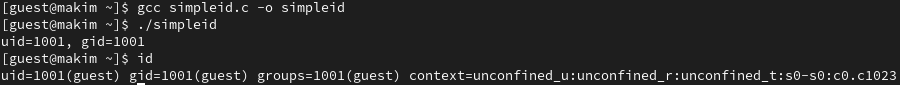
## 4.1 Создание программы

1. Создадим программу simpleid.c от имени пользователя guest командой nano simpleid.c (рис. [1](#fig:01)).

* #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;  
  }

* 
* Figure 1: Программа simpleid.c

1. Скомплилируем и выполним программу. Сравним полученный результат с выполнением системной команды id: информация совпадает (рис. [2](#fig:02)).

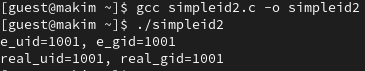
* 
* Figure 2: Компиляция и выполнение программы simpleid.c

1. Усложним программу, добавив вывод действительных идентификаторов. Назовем программу simpleid2.c (рис. [3](#fig:03)).

* #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;  
  }

* 
* Figure 3: Программа simpleid2.c

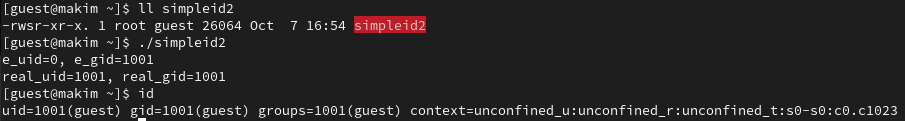
1. Скомпилируем и выполним программу (рис. [4](#fig:04)).

* 
* Figure 4: Компиляция и выполнение программы simpleid2.c

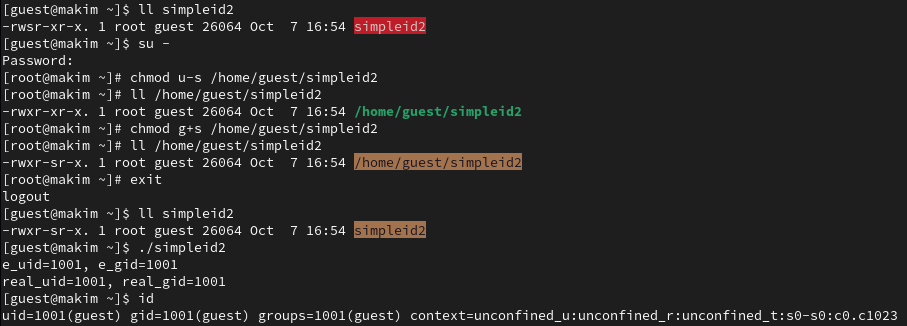
1. От имени суперпользователя выполните команды: chown root:guest /home/guest/simpleid2 - изменяем владельца файла simpleid2 с guest на root с указанием группы; chmod u+s /home/guest/simpleid2 - добавляем на файл SetUID-бит (рис. [5](#fig:05)).

* Figure 5: Изменение владельца файла и добавление SetUID-бита
* Figure 5: Изменение владельца файла и добавление SetUID-бита

1. Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой ll simpleid2. Запустим simpleid2 и id командами: ./simpleid2 и id. Сравним результаты: e\_uid у нас изменилось (рис. [6](#fig:06)).

* 
* Figure 6: Сравнение ./simpleid2 и id при установленном SetUID-бите

1. Проделаем аналогичные действия относительно SetGID-бита. Как видим, e\_uid у нас снова изменился (рис. [7](#fig:65)).

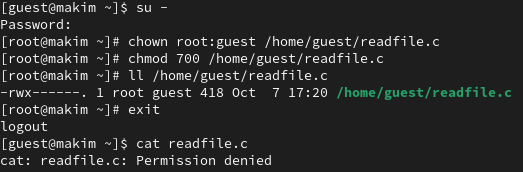
* 
* Figure 7: Сравнение ./simpleid2 и id при установленном SetGID-бите

1. Создадим программу readfile.c (рис. [8](#fig:07)).

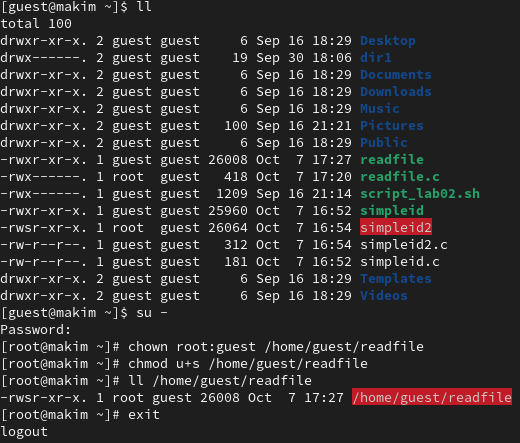
* #include <fcntl.h>  
  #include <stdio.h>  
  #include <sys/stat.h>  
  #include <sys/types.h>  
  #include <unistd.h>  
    
    
  int main (int argc, char\* argv[])  
  {  
   unsigned char buffer[16];  
   size\_t bytes\_read;  
   int i;  
   int fd = open (argv[1], O\_RDONLY);  
   do  
   {  
   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;  
  }

* 
* Figure 8: Программа readfile.c

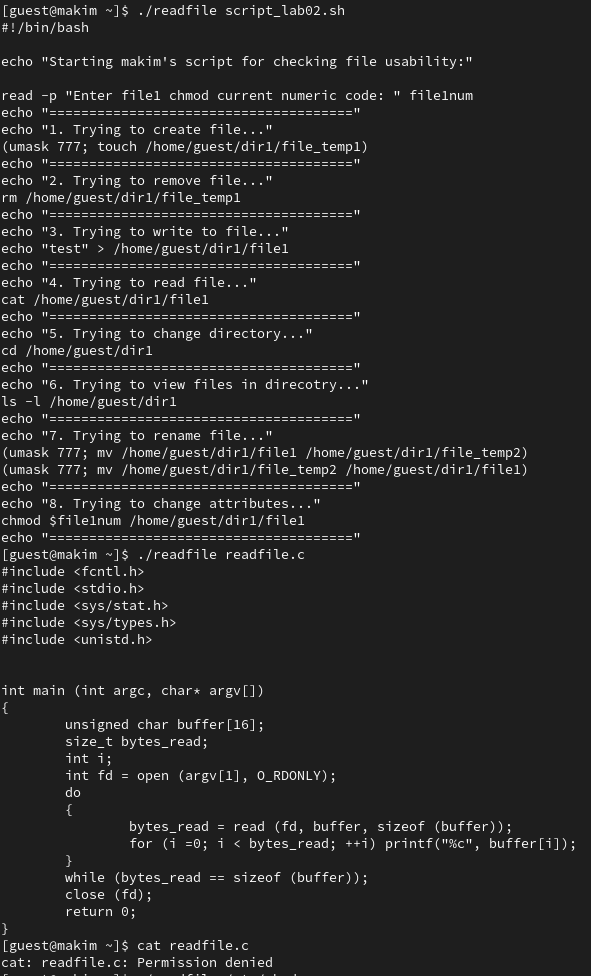
1. Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог. Проверим, что пользователь guest не может прочитать файл readfile.c. (рис. [9](#fig:08)).

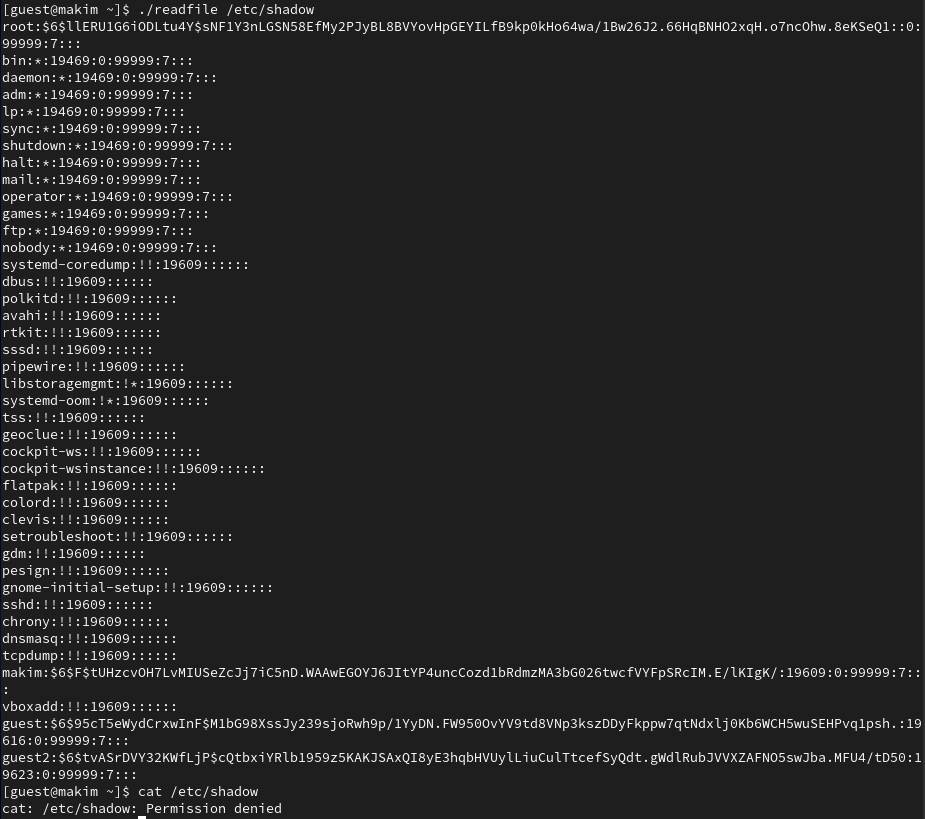
* 
* Figure 9: Смена владельца и прав файла readfile.c

1. Сменим у программы readfile владельца и установим SetUD-бит (рис. [10](#fig:09)).

* 
* Figure 10: Смена владельца и прав файла readfile

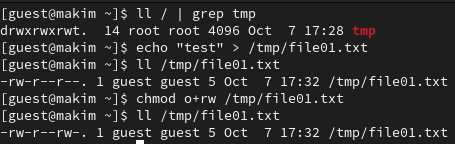
1. Проверим, может ли программа прочитать мой собственный файл, файл readfile.c, файл /etc/shadow. Как мы видим, программа может это сделать, так как владелец данной программы - суперпользователь (root) и установлен SetUD-бит. Соответственно, запуск программы осуществляется с правами пользователя - root. Поэтому мы имеем доступ к файлам, недоступным для пользователя guest (рис. [11](#fig:10), [12](#fig:11)).

* 
* Figure 11: Чтение файлов при помощи программы readfile. 1

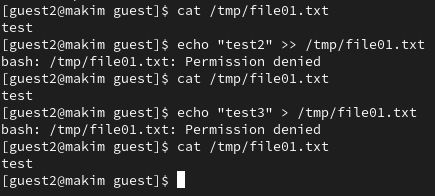
* 
* Figure 12: Чтение файлов при помощи программы readfile. 2

## 4.2 Исследование Sticky-бита

1. Выясним, установлен ли атрибут Sticky на директории /tmp, для чего выполним команду ll / | grep tmp. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test командой: echo "test" > /tmp/file01.txt. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные» командами: ll /tmp/file01.txt, chmod o+rw /tmp/file01.txt, ll /tmp/file01.txt (рис. [13](#fig:12)).

* 
* Figure 13: Просмотр атрибутов директории /tmp. Изменение атрибутов файла /tmp/file01.txt

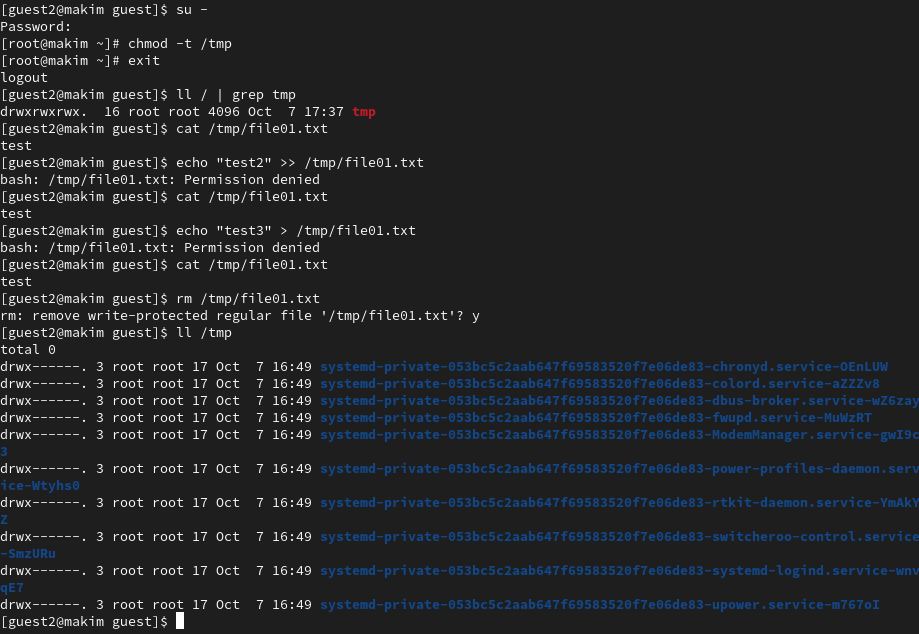
1. От пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2 командой echo "test2" >> /tmp/file01.txt. Выполнить данную операцию нельзя. От пользователя guest2 попробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo "test3" > /tmp/file01.txt. Выполнить данную операцию нельзя. (рис. [14](#fig:13)).

* 
* Figure 14: Попытка изменть файл /tmp/file01.txt от имени пользователя guest2

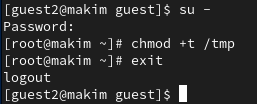
1. От пользователя guest2 попробуем удалить файл /tmp/file01.txt rm /tmp/fileOl.txt. Выполнить данную операцию нельзя. (рис. [15](#fig:14)).

* Figure 15: Попытка удалить файл /tmp/file01.txt от имени пользователя guest2
* Figure 15: Попытка удалить файл /tmp/file01.txt от имени пользователя guest2

1. Повысим свои права до суперпользователя и выполним команду, снимающую атрибут t с директории /tmp: chmod -t /tmp. Покинем режим суперпользователя. Проверим, снят ли атрибут t у директории /tmp. Повторим предыдущие шаги: теперь мы имеем возможность удалить файл /tmp/file01.txt (рис. [16](#fig:15)).

* 
* Figure 16: Попытка изменить и удалить файл /tmp/file01.txt от имени пользователя guest2 при отсутствии Sticky-бита на директории /tmp

1. Вернем обратно законный атрибут t для директории /tmp (рис. [17](#fig:16)).

* 
* Figure 17: Возвращение Sticky-бита директории /tmp

# 5 Анализ результатов

Работа выполнена без каких-либо проблем. Работа с терминалом ОС Rocky Linux в данном случае нареканий не вызвала. Также порадовало наличие вкладок в терминале «из коробки».

# 6 Выводы

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

# Список литературы

1. Терминал Linux [Электронный ресурс]. URL: <{https://www.reg.ru/blog/linux-shpargalka-komandy-terminala-dlya-novichkov/}>.

2. Права доступа [Электронный ресурс]. URL: <https://codechick.io/tutorials/unix-linux/unix-linux-permissions>.

3. Дополнительные атрибуты файлов [Электронный ресурс]. Enchanted Technology, 2022. URL: <https://wiki.enchtex.info/doc/linux_file_attributes>.

4. Использование SETUID, SETGID и Sticky bit для расширенной настройки прав доступа в операционных системах Linux [Электронный ресурс]. RuVDS, 2021. URL: <https://ruvds.com/ru/helpcenter/suid-sgid-sticky-bit-linux/>.