Отчет по лабораторной работе №5

Дисциплина: Основы информационной безопасности

Иванов Сергей Владимирович

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

Создаем программу simpleid.c (рис. 1)

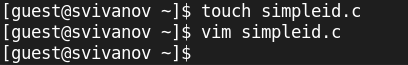


Рис. 1: Создание программы

Содержимое программы (рис. 2).

#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;  
}

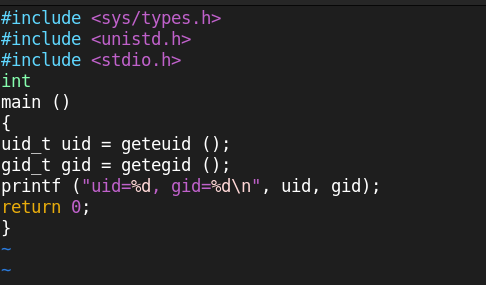


Рис. 2: Содержимое программы

Скомпилируем программу и выполним её (рис. 3).

Компиляция и выаполнение

Рис. 3: Компиляция и выаполнение

Выполним системную программу id. Сравним полученный результат с данными предыдущего пункта и видим что они совпадают (рис. 4).

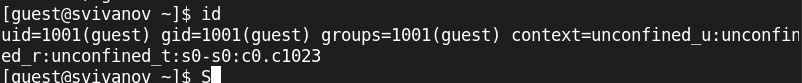


Рис. 4: Системный id

Усложним программу и назевем ее simpleid2.c (рис. 5).

#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;  
}

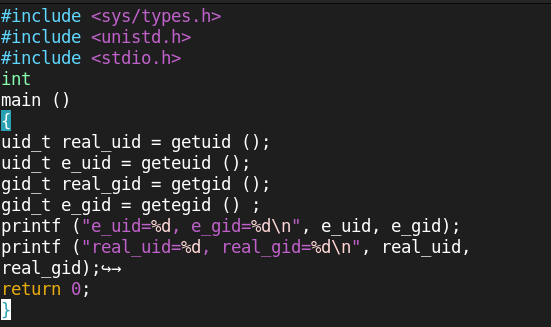


Рис. 5: Усложнение программы

Скомпилируем и запустим программу (рис. 6).

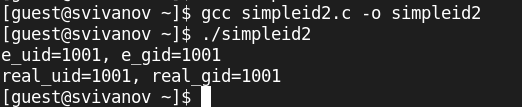


Рис. 6: Компиляция и запуск

От имени суперпользователя выполним команды: chown root:guest /home/guest/simpleid2 chmod u+s /home/guest/simpleid2. (рис. 7).



Рис. 7: Смена владельца директории и прав доступа файла

Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2: ls -l simpleid2. (рис. 8).

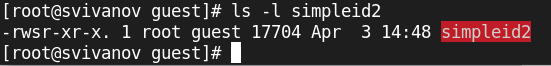


Рис. 8: Проверка смены атрибутов

Запустим simpleid2 и id: ./simpleid2 id Видим что вывод id более подробный (рис. 9).

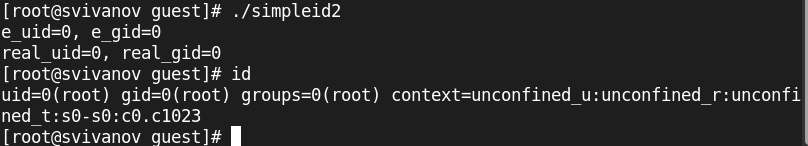


Рис. 9: Запуск программы и id

Создание и компиляция пррграммы readfile.c (рис. 10).

#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;  
}

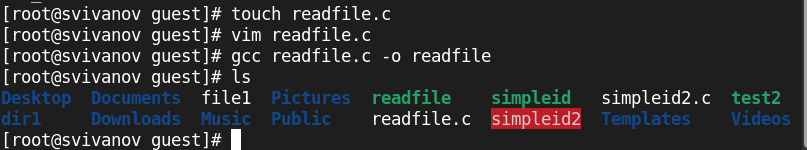


Рис. 10: Создание и компиляция программы

Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь мог прочитать его, a guest не мог. (рис. 11).

Смена прав и владельца readfile.c

Рис. 11: Смена прав и владельца readfile.c

Проверим, что пользователь guest не может прочитать файл readfile.c (рис. 12).

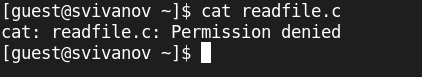


Рис. 12: Проверка чтения файла

Проверим, может ли программа readfile прочитать файл /etc/shadow? Не может (рис. 13).

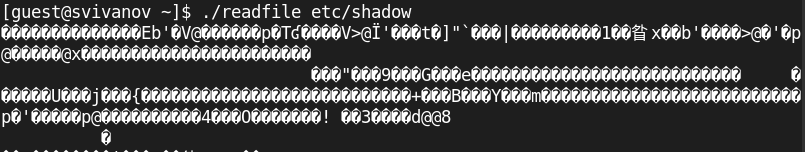


Рис. 13: Проверка чтения файла программой

Выясним, установлен ли атрибут Sticky на директории /tmp, для чего выполним команду ls -l / | grep tmp (рис. 14)

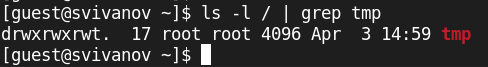


Рис. 14: Проверка атрибута Sticky

От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test: echo “test” > /tmp/file01.txt Просмотрим атрибуты у только что созданного файла и разрешим 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. 15)

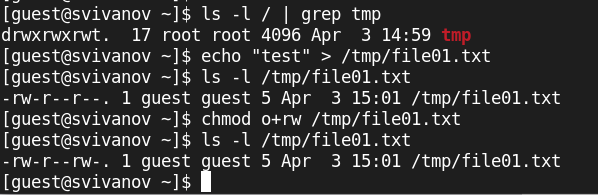


Рис. 15: Создание файла, изменение прав доступа

От пользователя guest2 попробуем прочитать файл /tmp/file01.txt, дозаписать слово test2: cat /tmp/file01.txt echo “test2” > /tmp/file01.txt Прочитать удалось, а записать нет (рис. 16)

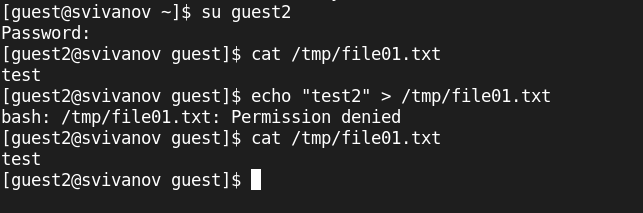


Рис. 16: Попытка чтения и записи

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 Операцию выполнить не удалось (рис. 17)



Рис. 17: Попытка записи

От пользователя guest2 попробуем удалить файл /tmp/file01.txt командой rm /tmp/fileOl.txt Не удалось удалить файл (рис. 18)

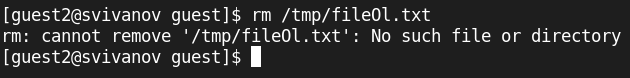


Рис. 18: Попытка удаления

Повысим свои права до суперпользователя и выполним команду, снимающую атрибут t с директории: chmod -t /tmp Покинем режим суперпользователя командой exit (рис. 19)

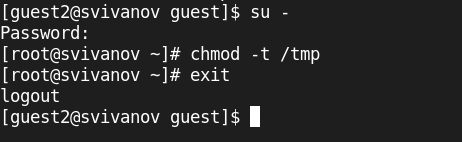


Рис. 19: Снимаем атрибут Sticky

От пользователя guest2 проверим, что атрибута t у директории /tmp нет: ls -l / | grep tmp (рис. 20)

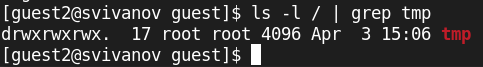


Рис. 20: Проверим снятие атрибута

Повторите предыдущие шаги. Записать в файл не получилось, но теперь стало доступно удаление. (рис. 21)



Рис. 21: Проверка предыдущих шагов

Повысим свои права до суперпользователя и вернём атрибут t на директорию /tmp (рис. 22)

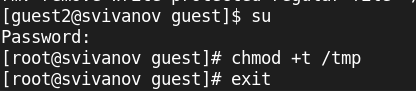


Рис. 22: Возвращение атрибута t

# 3 Вывод

В ходе работы были изучены механизмов изменения идентификаторов. Получены практических навыков работы с дополнительными атрибутами. Рассмотрены работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.