Tarea 1

Stevens Lira Eduardo Orlando

Sistemas Operativos

Grupo: 6

Fecha de entrega: 22 de Febrero de 2017

1. El sistema operativo utilizado fue GNU/Linux
2. Para obtener la traza se ocupo el programa strace, en la terminal de GNU/Linux.
3. El programa objetivo identificaba si un número primo, al ingresar cualquier número entero positivo. Fue desarrollado en lenguaje C.

El código del programa objetivo fue el siguiente:

#include <stdio.h>

int main()

{

int n, i, t = 0;

printf("Escribir un número entero positivo");

scanf("%d",&n);

for(i=2; i<=n/2; ++i)

{

if(n%i==0)

{

t=1;

break;

}

}

if (t==0)

printf("%d es un número primo",n);

else

printf("%d no es un número primo",n);

return 0;

}

Utilicé este programa debido a su simplicidad, lo cual me llevó a pensar que las llamadas del sistema que se generarían, serían fáciles de identificar y saber cual sería su función.

1. Las líneas consecutivas de las llamadas al sistema que elegí fueron:

**Ejecuta el programa entre paréntesis el cual es apuntado por nombre de archivo**

execve("/usr/bin/gcc", ["gcc", "sistop1.c", "-o", "prog2"], [/\* 35 vars \*/]) = 0

**Cambia la localidad del final del segmento de datos del proceso.**

brk(0) = 0x877b000

**Revisa si el proceso llamado puede acceder a la ubicación del archivo.**

access("/etc/ld.so.nohwcap", F\_OK) = -1 ENOENT (No such file or directory)

**Crea un nuevo mapeo en el espacio virtual de la localidad del proceso llamado.**

mmap2(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xb777a000

**Revisa si el proceso llamado puede acceder a la ubicación del archivo.**

access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)

**Asigna un número entero positivo que identifica un archivo para consecuentes llamadas al sistema.**

open("/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

**Otorga información acerca del archivo.**

fstat64(3, {st\_mode=S\_IFREG|0644, st\_size=83101, ...}) = 0

**Crea un nuevo mapeo en el espacio virtual de la localidad del proceso llamado.**

mmap2(NULL, 83101, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0xb7765000

**Desubica el numero descriptor del archivo obtenido en llamadas al sistema anteriores.**

close(3) = 0

**Revisa si el proceso llamado puede acceder a la ubicación del archivo.**

access("/etc/ld.so.nohwcap", F\_OK) = -1 ENOENT (No such file or directory)

**Asigna un número entero positivo que identifica un archivo para consecuentes llamadas al sistema.**

open("/lib/i386-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

**Como resultado da el número de bytes contados por el número obtenido por open() en el buffer.**

read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\220\226\1\0004\0\0\0"..., 512) = 512

**Otorga información acerca del archivo, se sabe que tiene éxito la función si regresa cero como es en este caso.**

fstat64(3, {st\_mode=S\_IFREG|0755, st\_size=1730024, ...}) = 0

**Crea un nuevo mapeo en el espacio virtual de la localidad del proceso llamado, esta función regresa un apuntador al área mapeada.**

mmap2(NULL, 1739484, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x9d7000

**Crea un nuevo mapeo en el espacio virtual de la localidad del proceso llamado, esta función regresa un apuntador al área mapeada.**

mmap2(0xb7a000, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a3) = 0xb7a000 Crea un **Nuevo mapeo en el espacio virtual de la localidad del proceso llamado, esta función regresa un apuntador al área mapeada.**

mmap2(0xb7d000, 10972, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0xb7d000

**Desubica el numero descriptor del archivo obtenido en llamadas al sistema anteriores.**

close(3) = 0

**Crea un nuevo mapeo en el espacio virtual de la localidad del proceso llamado, regresa un apuntador al área mapeada.**

mmap2(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xb7764000

**Sitúa una entrada en el arreglo del almacenamiento global del hilo actual, cuando es exitoso devuelve un cero, como es en este caso.**

set\_thread\_area({entry\_number:-1 -> 6, base\_addr:0xb7764900, limit:1048575, seg\_32bit:1, contents:0, read\_exec\_only:0, limit\_in\_pages:1, seg\_not\_present:0, useable:1}) = 0

**Cambia la protección de la llamada a las páginas de la memoria del proceso.**

mprotect(0xb7a000, 8192, PROT\_READ) = 0