Tarea 1: Depuración por *Trazas*

Cruz Santos Isaac

Sistema Operativo Usado:

Linux Ubuntu 16.04

Programa Empleado para Obtener la Traza:

Strace

Programa objetivo a Trazar:

Cp

*Escogí este programa por que estaba interesado en saber como se copian los archivos en un sistema operativo, más allá del modo grafico que siempre he visto en Windows.

scr3amind@scr3amind:~/Pictures/Tarea1$ strace cp ejemplo ejemplo1

execve("/bin/cp", ["cp", "ejemplo", "ejemplo1"], [/* 74 vars */]) = 0

execve: Ejecuta el programa indicado en este caso es “cp” que copia un archivo a otro; el archivo “ejemplo” a “ejemplo1”.

brk(NULL) = 0xf47000

brk establece el final del segmento de datos en este caso en la localidad 0xf47000

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

Access checa los permisos hacia un archivo

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f40b4a69000

mmap crea un area de memoria reservada en la localidad 0x7f40b4a69000

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

open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

fstat(3, {st_mode=S_IFREG|0644, st_size=104764, ...}) = 0

Fstat devuelve el estado de un archivo al que apunta.

mmap(NULL, 104764, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f40b4a4f000

close(3) = 0

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

open("/lib/x86_64-linux-gnu/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\260Z\0\0\0\0\0\0"..., 832) = 832

Read lee el archive a copiar,y al parecer son 832 caracteres

fstat(3, {st_mode=S_IFREG|0644, st_size=130224, ...}) = 0

mmap(NULL, 2234080, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f40b4624000

mprotect(0x7f40b4643000, 2093056, PROT_NONE) = 0

mprotect controla la forma en que la seccion de memoria puede ser accedida en este caso PROT_NONE,que significa que la memoria no puede ser accedida de ninguna forma.

mmap(0x7f40b4842000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e000) = 0x7f40b4842000

mmap(0x7f40b4844000, 5856, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f40b4844000

close(3) = 0

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

open("/lib/x86_64-linux-gnu/libacl.so.1", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\20\34\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0644, st_size=31232, ...}) = 0

mmap(NULL, 2126336, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f40b441c000

mprotect(0x7f40b4423000, 2093056, PROT_NONE) = 0

mmap(0x7f40b4622000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6000) = 0x7f40b4622000

close(3)   = 0

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

open("/lib/x86_64-linux-gnu/libattr.so.1", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300\20\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0644, st_size=18624, ...}) = 0

mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f40b4a4e000

El programa abre y protege nuevas secciones de memoria para copiar todo el archivo .

Resultados y Observaciones

Al tener poca experiencia en sistemas UNIX esta tarea fue muy didáctica, me ayudo a conocer y a usar distintos comandos en la terminal y a interesarme por su funcionamiento. Ver el sistema mas allá de la interfaz grafica para ver como funcionan realmente sus “engranes”.

Lo mas cercano a la tarea que hice antes fue depurar en turbo debugger, en mi curso de estructura y programación de computadoras, donde podíamos notar los cambios en registro y banderas, era una depuración por trazas también, a diferencia de ese tipo de depuración ,Strace vuelve las cosas un poco mas complejas al afectar varias áreas del SO y archivos del mismo. Abrir y cerrar secciones de memoria, protegerlas y cambiar su tipo de acceso.

El funcionamiento del comando “cp” es el “copy/paste” que estamos acostumbrados en nuestra interfaz grafica de windows, lee el archivo de origen, abre una nueva sección en la memoria y copia el contenido del archivo a la nueva área de memoria para después cerrarla.