CWE Challenge - Input1

Michael Mendoza

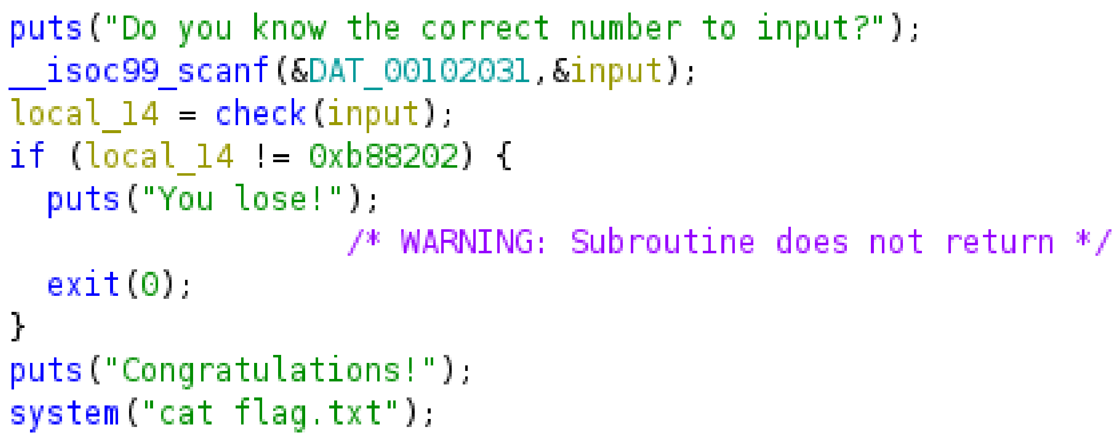
2023-01-21

Contents

Information Gathering

Ghidra

We can decompile the program using Ghidra and take a look at the main function.



```
puts("Do you know the correct number to input?");
__isoc99_scanf(&DAT_00102031,&input);
local_14 = check(input);
if (local_14 != 0xb88202) {
  puts("You lose!");
                    /* WARNING: Subroutine does not return */
  exit(0);
}
puts("Congratulations!");
system("cat flag.txt");
```

Figure 1: Main Function

We can see that after reading in the users input, its checked against 0xb88202, or 193480725 in base 10. If the user did not enter this number, then the program will exit. If it passes this check, a system call to cat the flag.txt file is called.

Creating the Exploit

To create the exploit, we will remotely connect using pwntools and then send the number needed to bypass the check.

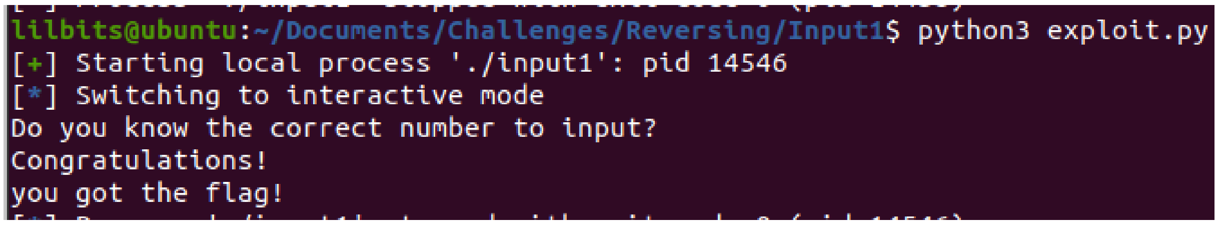
Python Script

```python
from pwn import *

target = remote('cweaccessionsctf.com', 1380)

target.sendline(b'193480725')
target.interactive()
```

Flag



```
lilbits@ubuntu:~/Documents/Challenges/Reversing/Input1$ python3 exploit.py
[+] Starting local process './input1': pid 14546
[*] Switching to interactive mode
Do you know the correct number to input?
Congratulations!
you got the flag!
```

Figure 2: Flag

And it worked!

Conclusion

Knowing the tools available to me to be able to decompile code is useful. In this case I used Ghidra, but using GDB or any other debugger and stepping through the problem would have been just as useful.