Computer Security Hw0x07 Writeup

- Realname: 胡安鳳
- ID on course web: alfons0329
- Student ID: R08922024

tags: Computer Security NTU CS CS CTF Writeup

casino++ (pwn)

Please see `solve.py` for more datails of exploitation code corresponding to the step in this writeup(both of step are the same).

Step 1 ~ 2: Control the flow of casino

Overwrite the puts function with the starting address of casino to make it run again such that we may futher exploit `read_int` where `idx` can be the only place for overflow.

- Note: Due to the 4bytes(32 bits) length limitation of integer, we can only write 32 bits per-try. Therefore, we need to first lose the game, write the address of casino in `puts@got`, and padding the rest 32 bits of zero in front of it.
- As in little-endian, `low{<casino>, 00000000}high` will be load as `0x00000000<casino>`

Step 3 ~ 4: Change from `srand(seed)` to `printf(libc)`

- Figure out the libc_offset

```
$ readelf -s libc-2.23.so | grep 'libc_start'
  2203: 0000000000021ab0   446 FUNC    GLOBAL DEFAULT   13 __libc_start_main@@GLIBC_2.2.5
```

- srand --> printf and seed --> libc_offset, with method similar to above.

```
gef➤  x/10gx 0x602040
0x602040:       0x0000000000400700
gef➤  x/w 0x602100
0x602100 <seed>:        0x00601ff0
```

  - Note: The payload has to be resolved printf, so we should write `0x400700` rather than `0x602030` which is not the proper path for resolving.

- And here we have the ASLR'd address of `libc`.

```
[*] Paused (press any to continue)
finished sending number and payload for step  4:
('get ASLR libc_base_addr: ', '0x7f0943e9c000')
```

```
37      in printf.c
gef➤  print $rip
$1 = (void (*)()) 0x7f0943f00f26 <__printf+166>
gef➤
```

Step 5 ~ 6: Change from `atoi(buf)` to `system('bin/sh')`

- It is harder to make `srand(seed)` to `system('/bin/sh')` since the int array of guess can only write 4 bytes at once, but in amd-64 we need 8 bytes at once, which is harder than making `atoi(buf)` to `system('/bin/sh')`.
- Also, the read_int() function provides more space to write system shell address!

```
int read_int(){
    char buf[0x10];
    __read_chk( 0 , buf , 0xf , 0x10 );
    return atoi( buf );
}
```

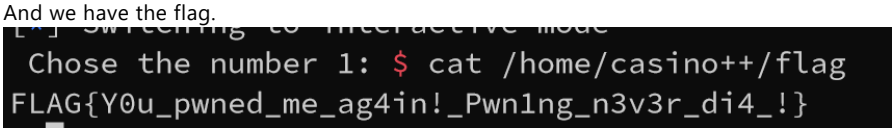
- Finally after `atoi` (0x602058) has been overwritten with `system`.

```
('get ASLR libc_base_addr: ', '0x7f0943e9c000')
('ASLR system_addr', '0x7f0943eeb440')
finished sending number and payload for step  5:
```

```
0x00000000004004ac0 in casino ()
gef➤  x/g 0x602058
0x602058:       0x00007f0943eeb440
```

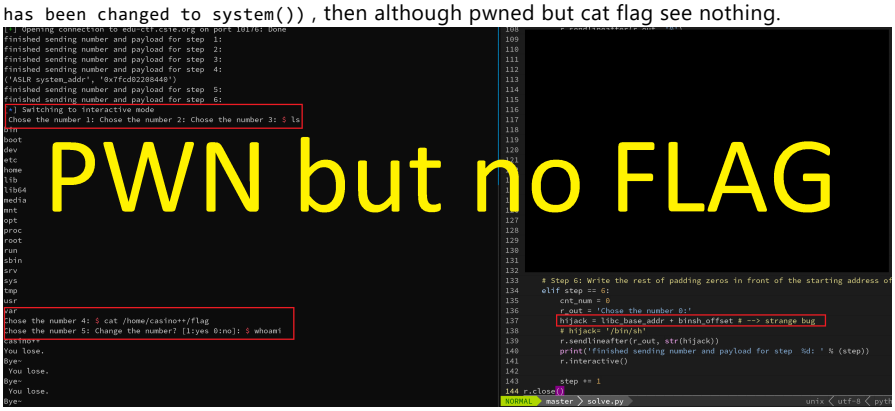
. Once `read_int()` has been invoked again, write '/bin/sh' to the buffer, and pwned!

- And we have the flag.

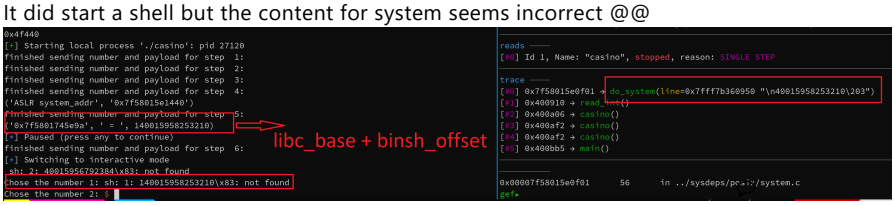


Pitfall bug in the final step

- And I have encountered a strange bug that if `<addr of /bin/sh>` is passed to `atoi(which has been changed to system())`, then although pwned but cat flag see nothing.



It did start a shell but the content for system seems incorrect @@



> Explanation from author, TA yuawn:

- Switch back to `'/bin/sh'` will be fine.