# hw1

### 0410788 CHING-WEN CHENG

### September 29, 2017

# Contents

# 1 Homework1

## 1.1 strings [50]

Use strings to list all printable character in files.

```
1    strings ./strings | grep "FLAG{"
```

```
FLAG{__flag_in_the_file}
```

## 1.2 strace [50]

Use strace to trace system calls.

```
1    strace -s 40 -e write ./strace 2>&1
```

```
1    write(2, "FLAG{____yaaaa_flag_in_the_stack___}", 36) = -1 EBADF (Bad file descriptor)
2    write(1, "find the flag in system call!\n", 30find the flag in system call!
3    ) = 30
4    +++ exited with 0 +++
```

## 1.3 patching [50]

First, we run the program.

```
1    ./patching
```

```
1    Value = 0x376c8
2    Go patching the value to 0x00023333
```

We can see that we need to patch `Value` to 0x00023333. Open `patching` with `emacs` in `hexl-mode`, find `c8 76 03 00`, little endian of 0x376c8, patch it to `33 33 02 00`, then run the program again.

```
1   ./patched
```

```
1   Value = 0x23333
2   FLAG{oa11TH8OwfMEs6ZflBhGF4btUcS1Ds9y}
```

## 1.4   pwntools [50]

Connect to the host, we can see that it is kind of "guess the number" game. We need to guess the correct number from 1 to 50000000, so write a simple binary search.

### 1.4.1   Code

```python
1   from pwn import *
2   M = 50000000
3   m = 1
4   def bs(x):
5       global r, M, m
6       r.recvuntil("= ")
7       r.sendline(str(x))
8       L = r.recvline()
9       l = L.split()[-1]
10      if l == b'big':
11          M = x
12          bs((M+m)//2)
13      elif l == b'small':
14          m = x
15          bs((M+m)//2)
16      else:
17          print(L)
18          return L
19
20  with remote("csie.ctf.tw", 10123) as r:
21      bs((m+M)//2)
```

```
1   [x] Opening connection to csie.ctf.tw on port 10123
2   [x] Opening connection to csie.ctf.tw on port 10123: Trying 140.112.31.96
3   [+] Opening connection to csie.ctf.tw on port 10123: Done
4   b'Success~ FLAG{hO2Ooysbv4O5Lf1Fmdrt2QKts7buYzOJ}\n'
5   [*] Closed connection to csie.ctf.tw port 10123
```